AFIT/DS/ENS/97-03

SCHEDULING AND SEQUENCING ARRIVALS TO A STOCHASTIC SERVICE SYSTEM

DISSERTATION

Peter Maurice Vanden Bosch
Major, USAF

AFIT/DS/ENS/97-03

AFIT/DS/ENS/97-03

SCHEDULING AND SEQUENCING ARRIVALS TO A STOCHASTIC SERVICE SYSTEM

Peter Maurice Vanden Bosch, M.S., B.A.

Major, USAF

Approved:

Dennis C Dietz	30 Oct 97
Edward F. Mykytka	30 Oct 97
Richard D Deckro	30 Oct 97
[illegible]	30 Oct 97
[illegible]	30 Oct 97

Robert A. Calico, Jr. 3 Nov 97

Robert A. Calico, Jr.

Dean

AFIT/DS/ENS/97-03

SCHEDULING AND SEQUENCING ARRIVALS TO A STOCHASTIC SERVICE SYSTEM

DISSERTATION

Peter Maurice Vanden Bosch
Major, USAF

AFIT/DS/ENS/97-03

AFIT/DS/ENS/97-03

SCHEDULING AND SEQUENCING ARRIVALS TO A STOCHASTIC SERVICE SYSTEM

DISSERTATION

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Peter Maurice Vanden Bosch, M.S., B.A.
Major, USAF

December 16, 1997

Acknowledgements

A number of people contributed materially to this dissertation. Professors Pu Patrick Wang of the University of Alabama and Alan Lair and Mark Oxley, both of the mathematics department at the Air Force Institute of Technology freely gave me their time and advice, as did my committee members, for which I am deeply grateful. I especially thank Professor Dennis Dietz, Lt Col, USAF, for suggesting this problem and shepherding me through the research. The consummate generosity and professionalism of the entire Air Force Institute of Technology library staff improved this effort in innumerable ways. I thank my mother, who taught me the beauty of mathematics, and my father, who taught me the joy of the journey, both of which have served me well in this endeavor. Most of all, I thank my wife, Marilyn Howe, Major, USAF, for her unflagging support.

Peter Maurice Vanden Bosch

Table of Contents

List of Figures

List of Tables

Abstract

Optimization of scheduled arrival times to an appointment system is approached from the perspectives of both queueing and scheduling theory. The appointment system is modeled as a single-server, first-come-first-served, transient queue with independent, distinctly distributed service times and no-show rates. If a customer does show, it is assumed to be punctual. The cost of operating the appointment system is a convex combination of customers' waiting times and the server's overtime. While techniques for finding the optimal static and dynamic schedules of arrivals have been proposed by other researchers, they mainly have focused on identical customers and strictly punctual arrivals. This effort provides substantially more efficient solution methods, addresses a more general cost function, allows for no-shows and non-identical service distributions, and applies either when arrivals are constrained to lattice points or when they are unconstrained. Because customers are not indistinguishable, this effort also provides heuristics for determining optimal customer order. The proposed techniques apply to any piecewise convex, submodular function.

SCHEDULING AND SEQUENCING ARRIVALS TO A STOCHASTIC SERVICE SYSTEM

I. Introduction

1.1 Motivation

Appointment systems have flourished over the last century. With widespread availability of the telephone and other forms of improved communications has come the realization of efficiencies due to scheduling that we often take for granted. For example, no longer do most people simply appear at a doctor's office for routine medical problems and wait, which was the rule in many societies even as late as the 1960s [45, 69]. Instead, an appointed time of arrival is agreed upon, many times by phone, and the server makes an implied commitment to attend to the customer as soon thereafter as the service protocol permits. This is the situation today for numerous personal services. It is also the practice in many industrial settings, such as the scheduling of cargo ships at port facilities, the scheduling of part deliveries in just-in-time systems and the scheduling of customers at military testing and training facilities.

Customers benefit greatly from such an arrangement, since their waiting times (as measured from their scheduled arrivals) are usually both smaller and less variable under a scheduling system. Servers incur costs due to the creation and maintenance of the scheduling system, the loss of some flexibility in operation, and the potential for increased server idle time. On the other hand, servers benefit from requiring smaller facilities for holding queueing customers and from increased customer satisfaction. Since customers and servers in general have different goals, the scheduling systems that optimize their interests will in general differ.

The major goal of this dissertation is to establish methods of optimizing measures of performance that characterize the interests of all parties, given various circumstances. The interest of each customer is taken solely as the minimization of its expected waiting time. The interest of the server is to minimize overtime. The cost of the system is assumed to be a convex combination of the waiting times and overtime.

Such a formulation has particular value when both the customers and server belong to the same organization, as in the case of an Air Force aerial combat range or military dental clinic. For such cases, the distinctions between server cost and customer cost blur, and there is greater certainty in the unit costs for each entity. For instance, it is possible to put a precise cost on the waiting time of each patient, each medical provider, and on facility availability for a military medical clinic, since the government incurs a calculable cost for each. The quantitative importance of serving a maximal number of customers may still have to be left as a value judgment, however; for instance, how does one unambiguously determine the monetary value to the military of a bombing training mission when scheduling a range, or the cost to society of a sick patient who can not work and must receive unemployment benefits as a result of delayed medical care [35]?

More difficult are situations in which the respective costs are incurred by distinct organizations – say, the use of one of the U.S. Air Force's Air Combat Maneuvering Instrumentation (ACMI) range facilities by a foreign air force, or the use of a military medical facility by someone not on active duty. In such cases, subjective judgments must be applied, such as "a doctor's time is 37.5 times more valuable than the patients' " [7].

A secondary goal of this dissertation is to show the applicability of the solution techniques developed here to other problems. It will be shown that the cost function arising from this optimization problem is related in structure to a large class of

problems in physics and resource allocation, and as a result, the solution methods may profitably be applied to these problems.

With these aims in mind, a more formal description of the problem is considered next.

1.2 Problem Description

A (fixed) set of N customers is to be assigned arrival times to a single server. Each customer arrives precisely on time unless failing to show altogether. The probabilities for each customer showing are known. Service time probability density functions (PDFs) are known and are independent of each other, of the show probabilities, and of the scheduled arrival times. The cost of operating the system is defined to be a convex combination of the individual expected waiting times and the server overtime. Figure 1 depicts the situation. Here, τ_i is the scheduled arrival time of customer i, while χ_i and W_i are the service duration and waiting times and I_i is the idle time of the server that ends at τ_i, each for a particular realization of the schedule.

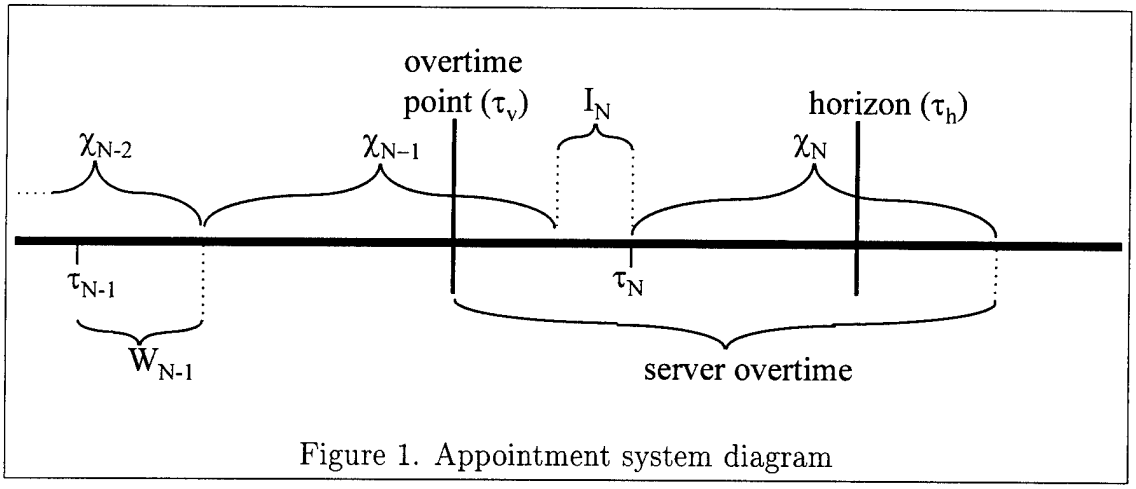


Figure 1. Appointment system diagram

Server overtime is defined as the time from some user-defined point, τ_v, to the time of completion for the last customer. Overtime is a generalization of constructs used in previous research. For example, by setting $\tau_v = 0$, overtime becomes the

total expected time the server must be available, which is the sum of the expected service time and the expected idle time. Since the expected service time is fixed, minimization of overtime in this case is equivalent to minimization of idle time.

Customers are constrained to arrive between 0 and the schedule horizon, τ_h. The first scheduled arrival time, τ_1, is fixed at zero, clearly its optimal value when lateness is not permitted.

A distinction will be made throughout between the schedule of arrivals and the sequence of arrivals.[1] To some extent, the sequence is determined by the schedule, but if two customers have identical arrival times, the cost may vary depending on who is served first. The approach will be to decompose the problem into two parts: finding the optimal schedule for a given sequence and finding the optimal sequence.

The cost of operation of a particular schedule/sequence of customers is defined as a convex combination of the expected customer waiting times and expected server overtime. With the inclusion of the constants τ_h and τ_v, this formulation is quite flexible. For example, by setting $\tau_v = 0$ and setting the schedule horizon to a sufficiently large value, overtime becomes the total time the server must remain in operation, and the cost function becomes one considered in several works [160, 97]. Another commonly used cost function can be obtained by setting $\tau_v = \tau_h + \Delta$, where Δ is the smallest schedule increment allowable (lattice size), in which case overtime assumes a more traditional meaning [145, 158].

Some special cases considered in this dissertation can be classified by the following characteristics:

- Service distribution. The development of the optimization algorithm will assume a general form for each customer's service time distribution, restricted

[1] Throughout this dissertation, the term "sequence" is used to refer to the sequence of planned customer arrivals, while "schedule" refers to the vector of planned arrival times for each customer. These definitions are natural to this problem, and they were advocated by earlier researchers in scheduling theory [22: p 450] [146: p 295]. However, they are a departure from current usage [130: pp 15-16].

only by the requirements that each be of bounded variation, have only positive support, and be independent of the other quantities involved. The cost evaluation implementation will restrict the form of the service distributions further, requiring them to be Coxian, with the addition of distinct probabilities of each customer requiring zero service (*i.e.*, failing to show for the appointment). The special cases of Erlang and iid (identically, independently distributed) services without no-shows will also be examined, since they lead to simplifications in the evaluation.

- Arrival constraints. In practice, the scheduling period is always bounded and frequently is of fixed length. The horizon does not restrict the end of service or the overtime point. Bounded, variable-length schedules are assumed unless otherwise stated. In the pure scheduling problem, the sequence is assumed already fixed, and so arrival times are constrained by $0 = \tau_1 \leq \tau_2 \cdots \leq \tau_{N-1} \leq \tau_N \leq \tau_h$.

- Server overtime. Unless otherwise stated, it will be assumed that $\tau_v = \tau_h$. This models a commonly observed appointment system, in which the server continues to be available to accept new customers right up until closing time.

- Arrival discipline – block *vs.* individual strategies. Scheduling strategies typically are classified as individual, block, or mixed. Individual strategies allow each arrival to be scheduled separately, while block strategies constrain arrivals to occur only at the beginning of each block of time. Blocks need not be the same size or contain the same number of customers. Mixed strategies typically consist of a set of simultaneous arrivals at the beginning of each block, with the remainder of the customers scheduled individually. This study seeks only the optimal individual schedule; since it is the least restrictive and encompasses the other cases, the optimal individual schedule cost is always less than the optimal costs for block or mixed schedules.

- Arrival discipline – continuous *vs.* lattice. In this dissertation, the case where arrival times are lattice (*i.e.*, restricted to occur only at discrete intervals) receives attention, particularly the case of evenly spaced lattices. While the cost under this condition can be obtained trivially from a continuous formulation of the problem, the regularly-spaced lattice arrival case is addressed separately for three reasons. First, in many cases it is realistic. Most appointment systems permit arrivals to be scheduled only at regular intervals; none allow for arrivals in continuous time, and few even schedule to the nearest minute. Second, it provides for simpler computation of the cost function. Last, the optimization of any function in lattice space requires special consideration, since the optimum in lattice space generally is not obtainable merely by rounding off the solution of the corresponding continuous case.

- Arrival discipline – no-shows and customer punctuality. The probability of a customer showing at the scheduled time and entering the queue is distinct and independent of queue size or other characteristics. Accounting for no-shows is of critical importance in many systems, since show rates can be as low as 70% [40, 66, 147]. Because a no-show may be considered as an infinitely late customer, it is reasonable to believe that accounting for no-shows is often of greater importance in modeling a system than accounting for lateness. The show probability is assumed to be 1.0 for each customer unless specified.

- Queueing discipline. It is assumed throughout that customers are served in the order they are scheduled. This is equivalent to first in, first out (FIFO), except in section 3.6, where customer lateness is incorporated into the model.

- Scheduling goal. The schedule may be dynamic or static. In the dynamic case, the schedule of future arrivals may be revised at any time, taking advantage of all information regarding past events and the current state of the system. The static case fixes the schedule prior to the start of service. The latter case will

be assumed. In some works the two problems are referred to a the short- and long-range versions of the problem.

- Optimization goal. It has been clear from past work in control of arrivals to a queue that the arrival time that optimizes expected cost from an individual's narrow point of view often differs from that selected in order to optimize the expected global cost [113]. Optimization is intended in the global sense in this work. There is a single objective; multicriteria optimization is examined only briefly.

1.3 Definition of Symbols, Terms, and Acronyms

Symbols and acronyms are defined at their first use throughout this document. For the reader's convenience, those that are used more than once are also defined below. Those symbols and acronyms that are used only once are not defined below. The lower-case symbols g, h, i, j, k, m, and n are used exclusively for indices and thus may have different meanings in different sections.

Kendall's notation for queues is used [80]. S(N) is used to denote a queue with N deterministic arrival times in which interarrival times may not be constant. The standard notations for sequencing problems will be seen to be inadequate for this sequencing/scheduling problem. When referring to other research, the notation for sequencing problems is that suggested by a number of authors and modified by Pinedo [130].

The following trademarks are mentioned: MATLAB (The Math Works, Inc.), Pentium (Intel), PowerStation (Microsoft), and SparcStation (Sun).

Table 1. Definitions of terms and variables

agreeable	A set of customers with deterministic services for which $\chi_i \leq \chi_j$ implies $c_i \geq c_j \ \forall i, j$ is said to possess agreeable weights.
block bidiagonal	Refers to a matrix that can be partitioned into blocks such that the $[i, j]$ block is nonzero only if $j = i$ or $j = i + 1$.
b_j	The probability of immediate completion of a Coxian service, given the j^{th} phase was just completed. A second subscript is added if not all customers have identical service distributions.
combination	A set of objects undistinguished by order. The permutations AAB, ABA, and BAA are all represented by the same combination, denoted by AAB in this research.
confluent	Refers to eigenvalues that are equal.
convex combination	$\sum_{j=1}^{N} \lambda_j x_j$ is a convex combination of the elements of x if λ is a vector of constants such that $\lambda_j \geq 0$ for all j and $\sum_{j=1}^{N} \lambda_j = 1$.
customer class	A subset of customers, each member of which is indistinguishable in terms of their probability of arriving on time, cost of service, and service time PDF, prior to schedule implementation.
c	Vector of unit costs of waiting. Also used as the coefficient of variation of a distribution.
c_{N+1}	Unit cost of server overtime.
$C(\tau, c)$	Cost of schedule of arrivals τ. The second argument is dropped if all unit costs of waiting are equal.
$\tilde{C}$	Optimal cost over all schedules.
χ_j	Service time of the j^{th} customer in a particular realization.
$X_{j,i}$	Expected remaining service time of the j^{th} arrival, given it is currently in its i^{th} Coxian stage of service. The initial subscript is dropped if all customers are identical.

Table 1. Definitions of terms and variables (continued)

dynamic problem	The problem of determining an optimal schedule sequence policy at any time during schedule implementation, given information on the realization of the stochastic variables (service time and whether the customer arrived) for each previous customer [129]. This is equivalent to closed loop control of arrivals [152].
distinct	Adjective denoting a set of probability PDFs that may not be all identical.
Δ	In the case of evenly-spaced lattice arrival times, the smallest possible positive time step.
E_r	Erlang service PDF with r phases or stages of service.
fathom	To prove to be suboptimal. A decision fathoms a set of alternative decisions if the cost incurred under the decision is lower than that incurred under each alternative.
FIFO	First-in, first-out service protocol.
f_j	PDF of service time for the j^{th} customer.
$\preceq$	When comparing two vectors, $x \preceq y$ iff $x_i \leq y_i \ \forall i$.
$\prec$	$x \prec y$ iff $x \preceq y$ and $x \neq y$.
γ_j	Probability of the j^{th} customer showing at the appointed time. The subscript is dropped if all customers have identical show rates. Also used as the skewness of a distribution.
idd	Independently, distinctly distributed.
iid	Independently, identically distributed.
I_j	The expected idle time the server incurs waiting for customer j.
int(x)	The greatest integer less than x.
join	$x \vee y = [\max(x_1, y_1), \max(x_2, y_2), \ldots]$ is the join of vectors x and y.
K	number of time slots in a lattice schedule in which a customer may be scheduled.

Table 1. Definitions of terms and variables (continued)

Maclaurin series	Special case of the Taylor series, expanded about zero.
mean residual life	The mean residual life, $L(t)$, is the expected remaining "life" of a process at t, given the process is still "alive" at t: $L(t) = E[\chi - t \mid \chi \geq t]$.
meet	$x \wedge y = [\min(x_1, y_1), \min(x_2, y_2), \ldots]$ is the meet of vectors x and y.
μ_j	The service rate of the j^{th} phase of a Coxian-r PDF. A second subscript is added if not all customers have identical service distributions.
NLP	Nonlinear program, a means of finding the minimum value of a nonlinear function or a function subject to nonlinear constraints.
norm	The norm of A, $\|A\|$, is defined as various metrics of a matrix (or vector). It provides some measure of the "size" of A.
N	Number of customers to be scheduled.
$N1$	Index of the first customer optimally scheduled at the horizon in the optimal schedule for a given sequence. This definition applies only for deterministic problems.
$\nu(j)$	The number of customers arriving in slot j.
$\Omega_{j,i}$	The expected waiting time of the j^{th} customer, given the system is in its i^{th} state (in a phase-type representation).
PDF	Probability density function.
preemption	Temporarily removal of the current customer from service in order to serve a higher priority customer.
priority	A ranking of customers by importance. In a priority queue without preemption, the highest priority customer present is served as soon as the current customer completes service.
P	Transition matrix, not including exit phase.
ϕ_i	The i^{th} noncentral moment of a distribution.

Table 1. Definitions of terms and variables (continued)

Φ_i	The i^{th} scaled, noncentral moment of a distribution.
Ψ_{N+1}	A row vector of $N+1$ elements, all equal to one.
Q	Transition matrix, including exit phase.
$\mathcal{Q}$	Denotes the set of all rational numbers.
realization	The value of a random variable in a particular trial.
r_i	The number of phases in the Coxian service distribution of the i^{th} customer. The subscript is dropped if all customers have identical service distributions. The variable is also used briefly in Chapter II to refer to release dates.
R	Right-shift operator, with accumulation in the last entry: if $x =$ [12345], $R(x) =$ [01239]. It is also defined as the matrix for which $R(x) = xR$.
$\Re$	Denotes the set of real numbers. $\Re^+$ denotes the positive real numbers, including zero.
scheduling	Determining arrival times.
sequencing	Determining the sequence of arrivals.
static problem	The problem of determining an optimal schedule sequence policy at any time during schedule implementation, in ignorance of the realization of the stochastic variables (service time and whether the customer arrived) for each previous customer. This is equivalent to determining the optimal schedule prior to schedule start. It is also referred to as open loop control of arrivals [152].
submodular	A function $f : \Re^n \rightarrow \Re$ is submodular on $\Re^n$ if $$f(x \wedge y) + f(x \vee y) \leq f(x) + f(y) \; \forall x, y$$ where $\wedge$ is the meet operation and $\vee$ is the join operation.
support	The set of numbers for which a PDF is nonzero is called the support of that PDF.

Table 1. Definitions of terms and variables (continued)

$S(N)$	Used by several references to denote deterministic arrival times to a queue – *e.g.*, $S(N)/\mathrm{G}/1$.
S_i	The i^{th} schedule being considered. Used at times in place of τ, since τ_i refers to the scheduled arrival time of the i^{th} customer.
$\hat{S}$	Optimal schedule of those constrained to a given lattice.
$\tilde{S}$	Optimal unconstrained schedule.
t	time, measured from the start of the scheduling period.
τ_h	The schedule horizon; the time interval within which customers may be scheduled.
τ_j	Arrival time of the j^{th} customer. τ_1 is taken to be zero unless otherwise specified.
τ_v	The time that overtime costs begin.
θ_j	Show vector. Defined as 1 if, in a particular realization, the j^{th} customer arrived at the appointed time, zero otherwise.
upper triangular	Denotes a matrix A in which $A_{i,j} = 0 \ \forall i, j : i > j$.
W_j	The expected waiting time of customer j.
WSEPT	Acronym for weighted shortest expected processing time. The customer sequence obtained by ordering customers from smallest to largest value of $\mathrm{E}(\chi_i)/c_i$ in stochastic sequencing problems.
WSPT	Acronym for weighted shortest processing time. The customer sequence obtained by ordering customers from smallest to largest value of χ_i/c_i in deterministic sequencing problems.
WSVPT	Acronym for weighted shortest variance of processing time. The customer sequence obtained by ordering customers from smallest to largest value of $\mathrm{VAR}(\chi_i)/c_i$ in stochastic sequencing problems.

1.4 Overview

A part of the problem this dissertation addresses – the scheduling of arrivals to an appointment system in order to minimize cost – is one that has been addressed in over 60 articles in the last 47 years. This dissertation builds on these earlier attempts to model appointment systems realistically and to develop efficient approaches to optimizing the schedule for a given sequence of arrivals. It provides substantial improvements in both these areas. In addition, this effort explores the effect of changing the sequence of customer arrivals, and it offers a heuristic algorithm to determine the optimal sequence. The importance of sequencing of arrivals to an appointment system has only been discussed once in the literature, and no optimization algorithm was proposed. The literature related to these problems is discussed in detail in Chapter II.

Chapter III considers the formulation and evaluation of the function representing the cost of a particular schedule and sequence of arrivals for a given system. It approximates customer services with Coxian distributions and employs a continuous Markov chain embedded at the customer arrival epochs to determine expected waiting times. Alternative approaches are developed for general distributions and Erlang distributions. A cost evaluation scheme that accounts for customer lateness is shown here, but lateness will not be addressed in subsequent sections. The nature of the cost function is considered, as a prelude to the optimization effort.

The optimization problem naturally presents itself as two sub-problems. Chapter IV addresses the determination of the optimal arrival times of each customer, given the sequence that the customers are scheduled to arrive. An algorithm to find the optimal lattice schedule is developed and proven analytically to be effective. It is based on the piecewise convexity and submodularity (*cf.* Section 2.5) of the cost function with respect to scheduled arrival times.

Chapter V addresses the determination of that optimal sequence in which customers are scheduled to arrive. The problem is demonstrated to be complex, with

optimal sequences appearing chaotic to the casual observer. Although the problem is suspected to be strongly NP-hard, a heuristic algorithm is shown empirically to perform effectively in determining the optimal sequence in polynomial time.

The last chapter is devoted to a discussion of the contributions of this dissertation to the research community. Applicability to various actual problems will be evaluated. Several goals for future research and possible approaches to those goals are offered.

The deterministic analogue to the problem of sequencing and scheduling arrivals to an appointment system is examined in Appendix A. While this problem is unrealistic in and of itself, the complexities seen in the stochastic solution have their root in this problem, and it is therefore worth considering.

Functions with the same piecewise convex and submodular structure as the one addressed here are ubiquitous and span a number of disciplines. Appendix B discusses the advantages and limitations of using the methods applied in Chapter IV to optimize such functions over a lattice.

Appendix C addresses the complexity of the optimization methods advocated in this dissertation. In particular, it is shown that the number of function evaluations required by the lattice scheduling algorithms is of linear order with respect to problem size for nearly all cases, making it superior to other optimization approaches beyond some problem size. The results of the fixed-lattice algorithm are compared to those of other methods and indicate the algorithm performs favorably at small problem sizes as well.

In an effort to gain more understanding of the nature of the problem, Appendix D examines the dependence of the optimal cost and schedule on a variety of factors, including service moments, show rates, and unit costs of waiting and overtime.

Appendix E describes a study performed for a medical clinic. Their appointment system was analyzed and the potential improvement attained from sched-

ule/sequence optimization was determined to be 67%. Although preliminary, this study is strong evidence that the approaches advocated here can be of immediate practical value.

Appendix F describes and analyzes approaches to finding Coxian distributions with the same moments as those of some empirical distribution. The tools developed include: a recursive approach to determining the moments of a Coxian distribution; completion of earlier researchers' work in determining the bounds of the Coxian-2 distribution in moment space; and the determination of a parsimonious set of Coxian parameters to match the first three moments of a given distribution. These tools were helpful to this research, since Coxian service distributions are assumed throughout.

Appendix G provides an analysis of several approaches to matrix exponentiation. Matrix exponentiation is necessary in this dissertation to finding the cost of a given schedule. Commonly-used algorithms such as those of Cayley and Hamilton, Jordan, and Parlett are shown to have numerical instabilities in this problem that preclude their use. No such problems are found with Padé or Maclaurin series methods when coupled with a scale-and-square algorithm.

II. Related Work

The following sources are divided into somewhat artificial categories. Articles in the heuristic appointments section emphasize attacking the problem from a practical standpoint by applying a heuristic and are most often formulated in terms of medical patient scheduling and sequencing. On the other hand, articles in the theoretical appointments section tend to start from a theoretical framework. However, the most important distinction is that, with few exceptions, researchers classified in one category appear to have little influence on the efforts of researchers in the other category. Figure 2 shows the ideological heritage for a representative selection of the 64 articles in the two sections, as determined by the articles they cite.

Articles in the control of queues section emphasize formulation of the problem in terms of control of arrivals and do not rely on any of the research in the other sections, although some of the researchers cited in the theoretical appointments section mention control of queues in passing. All three of these sections concentrate on single servers and identical customers, so most only address the scheduling problem. The fourth section addresses the sequencing problem. While the dividing lines between the four research areas are not always defined clearly, the distinctions are useful here.

Last, the literature related to optimization of submodular functions is reviewed, since the cost function discussed here will be shown to be submodular with respect to the arrival time vector, when the sequence of arrivals is fixed.

The first published considerations of the problem of scheduling arrivals found in this literature search were in 1951. In the discussion following Kendall's presentation on queueing theory in general, Herne briefly mentioned the scheduling of iron ore ships into English ports as a long-standing problem [80]. There was no discussion on how to formulate or solve the problem. In the same year, concerns regarding medical appointment scheduling (excessive waiting times, lateness of patients and

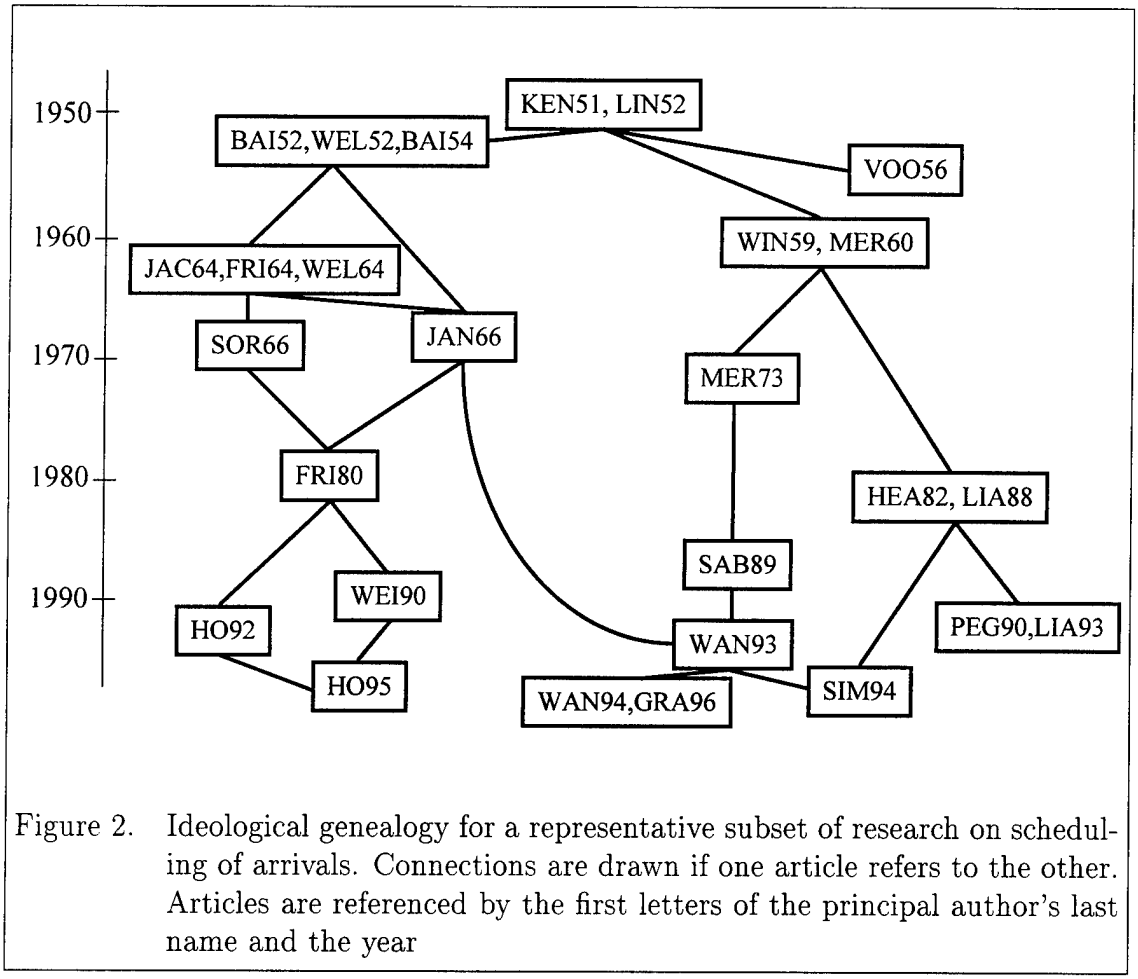


Figure 2. Ideological genealogy for a representative subset of research on scheduling of arrivals. Connections are drawn if one article refers to the other. Articles are referenced by the first letters of the principal author's last name and the year

staff, and the need to collect and analyze scheduling data) were addressed in the medical literature by Dale [30].

2.1 Heuristic Appointment Scheduling Literature.

Bailey suggested in 1952 that, in many medical clinics, the ratio of waiting time to service time was too high and could be addressed by application of queueing principles. In particular, he argued against the common practice of blocking – *i.e.*, scheduling a set number of patients at the beginning of a set of regularly-spaced intervals. While this practice, especially the then-common use of a single block for the entire day, minimizes the doctor's idle time, it also maximizes the patients' waiting time.[1] He recommended an individual scheduling scheme, in which patients are

scheduled individually at regular intervals. On the basis of earlier work, he fit truncated Pearson Type III curves to the service distributions of 50 practices. Using a Monte Carlo approach, he obtained waiting time distributions for a small number of patients. No recourse was made to steady-state approximations of expected waiting time. He discussed the trade-offs between reducing patients' waiting time and the doctor's idle time and suggested a reasonable target ratio between the two could be achieved with only a negligible increase in doctors' idle times [7, 8, 120].

Despite Bailey's emphasis on individual scheduling, he recommended partial blocking; an optimal schedule should have several patients arriving before the start of service. Given his assumption of punctual arrivals, this strategy would only serve to increase patient waiting time over that achieved for individual scheduling, without decreasing doctor idle time. He may have tacitly relaxed the punctuality assumption and was allowing for the possibility of late arrivals, in which case such a policy is sensible [7, 8].

Welch echoed many of Bailey's suggestions but emphasized the importance of punctuality in reducing queue size, both for patients and for doctors. He approached the subject with less rigor, proposing a mixed block-individual appointment scheme. In this scheme, interarrival times were all set equal to the average service time, and two patients arrived at the start of the day. There are several problems with this proposal. Equal interarrival times do not yield optimality in the transient case. Interarrival times should always be greater than the mean service time to prevent waiting time building over the day. Unless customers are likely to be very late, placing a second customer at the beginning of the day increases waiting time with

[1]The fact that appointment systems are a rather recent invention is evident from a comment in the 1955 Nuffield study, citing a 1932 British study: "They believed that any comprehensive system of appointment, giving each outpatient a separate time, was 'an impossible ideal'; but they thought nevertheless that in some of the special departments an appointment system for individual patients or groups of patients might be possible [120] ." A doctor's 1964 comment that, "One of the suggested improvements in general practice has been the introduction of an appointment system for the patients" implies that individual appointment systems were by no means ubiquitous even then [45].

no concomitant decrease in idle time. However, Welch believed these simplistic rules led to a reasonable, albeit suboptimal, solution [166]. Several articles discuss actual applications of this scheduling scheme [45, 69, 167].

Soriano extended Welch's work, comparing costs for block, individual, and mixed schedules. By appropriate choices of (constant) interarrival time and the initial block size in a mixed block-individual system, a reduction in waiting time of 50% was achieved in a hospital outpatient department. He also obtained steady-state waiting time distributions for various load factors in M/G/1 and $D/E_r/1$ systems [149, 150].

Computer simulations were the most common approach to the problem from 1964 to 1981. White and Pike concentrated on systems in which arrival times are not deterministic and services are identically and independently distributed (iid), not necessarily exponentially. They proposed a block system in which a day is divided into approximately 10 blocks and a fixed number of patients are scheduled at the beginning of each block. Average interarrival times were still set equal to the mean service time [168]. Fetter and Thompson studied effects of patient load, lateness of both patient and physician, and variations in interarrival times [38]. Katz developed a Monte Carlo model of hospital outpatient scheduling that took into account lateness, multiple physicians and their schedules, and lab scheduling. A candidate schedule could then be input to determine areas with excessive waiting times [79]. Several simulations were applied to the problem subsequently [9, 36, 51, 55, 59, 39, 59, 85, 86, 139].

Fries and Marathe considered a system in which arrival times are deterministic, services are iid exponential, and cost is a function of total patient waiting time, server idle time, and server overtime. They proposed a variable sized, multiple block system, in which the number of schedule blocks is predetermined, but their duration is the product of the mean service time and the number of patients scheduled to arrive at the beginning of that block. They proved that the cost is convex with respect to

the arrival time vector, which consists of the number of patients scheduled at the beginning of each slot. This is the first known proof of convexity of cost function for a type of finite-customer queue. The optimum schedule was then determined using dynamic programming [44]. Although arrival times were restricted, this is also the first theoretical consideration of a system in which the interarrival times were not necessarily equal. For this reason, the research should be classified with the theoretical appointment scheduling literature. However, it is discussed here because none of the researchers discussed in the next section cited it, while subsequent authors cited in this section were aware of it and used it.

Charnetski considered the problem of individually scheduling a fixed number of surgeons into a hospital operating suite during a fixed period. He approximated the cost of the surgeons' total waiting time and the cost of idle time of the suite (which includes both the facility and the dedicated operating room personnel) using a Monte Carlo simulation. Arrival times were deterministic, and service distributions were distinct truncated Gaussians. He found the optimal interarrival time which would balance the approximated costs of waiting and idling [21].

Weiss independently considered a nearly identical problem, solving it completely for two customers in the case of independently and distinctly distributed (idd) general service times and dynamic scheduling. He offered a heuristic for the case of scheduling more than two customers that was tantamount to one iteration of a cyclic coordinate search. His is the first published attempt to solve the problem of sequencing distinct surgical procedures [164]. The general literature regarding scheduling policies for operating rooms is extensive, but much is not pertinent to the problem at hand, since the objective is usually to balance waiting times, rather than to minimize them. The interested reader is referred to reviews in [132] and [102].

Ho and Lau compared a number of block scheduling schemes for the case of deterministic arrival times, but with possible no-shows, and either iid uniform or iid exponential service distributions. They cited previous, unpublished research showing

that distributions that differed only by third and higher moments yielded essentially identical costs. They optimized the cost with respect to block scheduling parameters, as well as the number of patients served. A number of scheduling strategies were considered, and a set of eight were identified as optimal under different parameters. Remarkably, one of these was Bailey's and Welch's scheme of 40 years earlier, in which two customers arrive before the service even begins, and the remaining interarrival times are set equal to the mean service time [63, 64, 65]. This work paralleled, but was independent of, simulation work performed much earlier by researchers at the Air Force Institute of Technology [12, 51, 59].

2.2 Theoretical Appointment Scheduling Literature.

More mathematical approaches to schedule optimization began with a brief discussion in 1956 in *Operations Research* of the problem of minimizing a combination of waiting and idle times for a steady-state system [159]. A similar problem was formulated and solved by Morse in relation to the scheduling of ships to a docking facility for the case of a steady-state M/M/1 system [111].

Jansson was first to obtain the optimal interarrival time of a steady-state D/M/1 queue, in 1966 [70]. In 1968, Grape extended Jansson's work to explore the rate of convergence of transient D/M/1 queues to steady-state. As might be expected, convergence is dependent on the queue size at the start of customer service and on the utilization (the ratio of expected service time to interarrival time). This is the first known consideration of scheduling arrivals to a transient queue [56].

Fries and Marathe investigated the scheduling of arrivals to a S(N)/M/1 queue under transient conditions, as discussed above [44]. The S(N) notation denotes deterministic arrival times with distinct interarrival times. They optimized variable-length block schedules in which block size was a fixed multiple of the number of customers in the block. Other researchers discussed in this section do not cite this work or the work of Grape.

Pegden and Rosenshine addressed the optimization of variable-length individual schedules [125, 127]. In this formulation, arrivals are scheduled in continuous time, services are iid exponential, cost is a linear convex function of expected waiting times and expected server availability (the overtime as measured from zero), and the scheduling horizon is not fixed. Their cost function is a special case of the cost function proposed in this dissertation. For this S(N)/M/1 problem, they were able to prove convexity for the cases of $N = 2$ and $N = 3$. They provided the exact solution of the optimization problem for $N = 2$ and solved numerically when $N = 3$. They formulated the cost function for larger values of N and chose a Hooke-Jeeves optimization because of the difficulty in obtaining derivative information. However, the convexity of the cost function in all cases was not resolved. Healy, Pegden, and Rosenshine later extended their work to consider two parallel servers ($S(N)/M/2$) for a small number of customers [60, 61].

Difficulties in the continuous-time formulation led Pegden and Rosenshine to restrain arrival times to fixed lattice points. As mentioned above, this formulation is more representative of most actual problems. They used dynamic programming to solve the dynamic version of the problem, in which the scheduling decisions for future arrivals are revised at each time interval. They suggested approaches for solution of the static problem as well. Convexity of the cost function for $N > 3$ was conjectured but not established [126].

Liao extended these results to the iid S(N)/E_r/1 case of lattice arrival times and a finite schedule horizon. He obtained the optimal dynamic schedules by a recursive scheme and then used these solutions as lower bounds to solve the static case by a branch-and-bound algorithm. The cost function was also formulated for multiple class and multiple server problems. While he proved convexity of the cost function for the dynamic version of the problem, convexity for the static version for $N > 3$ was still unresolved. Liao noted the applicability of his work to just-in-time inventory systems [95, 96, 97].

Simeoni proposed a different search procedure for the problem Liao proposed, implementing it for the case of iid Erlang services. This procedure modified the schedule by only one customer arrival at a time but in a way that fathomed large amounts of the solution space. The issue of convexity was still unresolved [145]. Vanden Bosch and Dietz showed that, while Simeoni's proof was incomplete, the method was sound and relied on the submodularity, rather than the convexity, of the cost function. They extended the algorithm to the optimal scheduling of arrivals with iid Erlang service distributions, and proved it was applicable to general service distributions if the cost could be evaluated [158]. Further generalizations of the algorithm are possible and will be discussed in Chapter IV.

Wang was the first to prove stochastic convexity of a cost function for the general S(N)/G/1 problem. His cost was a convex function of waiting times and server availability (the sum of the scheduling horizon and overtime), the arrival time space was continuous, and the scheduling horizon was not fixed or bounded. He addressed both the static and a dynamic scheduling problem for iid phase (PH) service distributions. He obtained the optimal solution by applying a gradient search to a series of differential equations in matrix form [160]. Subsequently, he addressed the applicability of the problem to just-in-time systems in which steady-state is not reached due to work stoppages [161]. Two works not yet published propose efficiencies in his approach and compare optimal transient schedules to the steady-state results of Jansson [57, 163].

Wang noted the substantial simplifications to obtaining expected waiting time that accrue using phase-type distributions. However, most of his results depend on placing the phase transition matrix in Jordan canonical form [160]. As will be shown in Appendix G, this can cause severe floating point errors in certain situations that are frequently encountered.

Several authors subsequently investigated the scheduling of arrivals at equal intervals, assuming that customers are allowed to arrive late according to some distri-

bution. Several authors approached the issue of lateness from a simulation approach [38, 55, 79, 168]. Winsten first considered an analytical approach, in which he modified a D/M/1queue to allow for lateness. The lateness distributions were iid and were general, with the restriction that customers were not allowed to change order. He determined steady-state measures of this system [171]. Mercer extended these results to multiple servers, bulk arrivals, and a more general staged service distribution. While he did not allow the j^{th} customer to arrive before the $(j-1)^{st}$, he did allow it to arrive at any time later. Again, steady-state measures were sought [104, 105]. Sabria and Daganzo examined a transient queue with scheduled arrivals (*i.e.*, an appointment system) in which customers are forced to balk if they do not arrive in the order scheduled. Lateness distributions and service distributions are iid and general. They obtained approximations to queue length and waiting time that, for the steady-state case, were in error by less than 10% [137].

Several researchers have considered similar deterministic scheduling problems. None directly apply to the deterministic analogue to the problem of scheduling arrivals to an appointment system, which is addressed in Appendix A. They are not reviewed here, but the interested reader is directed to several recent articles with good surveys [10, 11, 81, 88, 144].

2.3 Control of Queues Literature.

A number of the above researchers have suggested the scheduling of arrivals problem is connected to the control of arrivals to a queue. In these problems, decisions are made at various times whether to accept or reject entry of an arrival to the system. While none of the literature in this area was deemed applicable to the current problem, a brief examination of the literature is appropriate, if only to provide the reader with an understanding of why this approach was rejected.

Heyman was the first to consider a problem in the control of arrivals to a queue. He developed optimal policies for a M/G/1 queue whose output is controlled

by turning the server off and on [62]. Naor addressed the control of arrivals to a M/M/1 queue by imposing an entrance fee on arrivals and allowing each customer to decide whether to enter the queue. He obtained optimal strategies both for the case in which each customer exercises its own narrow self-interest and for the case in which each customer considers the expected cost to the customers as a group. Although these strategies both are of the form, "enter the queue if the cost is less than some fixed amount", they are not equivalent [113].

Johansen and Stidham considered a problem close to the one of interest. They controlled a transient GI/G/1 queue by accepting or rejecting arriving customers. A reward was accrued for every accepted customer, but a cost based on the waiting time was also incurred. Problems were considered in which the customers exercised self-interest and in which the global interest was optimized. One of the difficulties they encountered was again the question of convexity of the cost function [71].

Several sources provide good reviews of the subsequent research on control of arrivals to a queue [27, 33, 82, 148, 153]. These efforts may be classified by several characteristics.

- Control discipline. Control may be open-loop, in which case the current decision must be made in ignorance of past and current states of the system, or closed-loop. These optimal control strategies are equivalent to the optimal static and dynamic strategies, respectively.
- Queueing system. Researchers have addressed multiple servers, balking, reneging, multiple customer classes, networks – in short, the whole spectrum of queueing systems.
- Objective function. The objective may be to minimize average expected cost, cost variance, maximum expected cost, average expected queue length, or some other possibility. Costs may be customer-oriented or system-oriented. As

mentioned above, it must be specified whether the basis for customer decisions is self-interest or global interest.

- Equilibrium. The control of arrivals literature addresses only steady-state cases, with few exceptions [34, 71, 82].
- Control space. Control of arrivals generally has been pursued by modifying service rates, by modifying cost functions, or by routing arrivals to different servers.

Researchers are deterred when trying to apply these efforts in control of arrivals to the problem of scheduling arrival times to an appointment system for several reasons. First, few of the efforts in control of queues address transient queueing problems. Second, control in this problem is achieved by modifying the arrival times, a situation the control literature does not appear to address. Last, because the parameters in the cost functional are not time-dependent, the control formulation does not provide any benefit; it degenerates to an optimal design formulation, and nonlinear programming (NLP) techniques are directly applicable [27]. Thus, no advantage is accrued by formulating it as a control of arrivals problem. As a result, the problem will not be considered further in terms of control of arrivals.

2.4 Sequencing Literature.

In the case of identical customers, the problem can be approached strictly from a queueing theory standpoint. Stochastic scheduling theory is irrelevant, since it is concerned with the optimal sequencing of customer services. However, when the customers differ in cost coefficients, service distribution, or are otherwise distinguishable, the sequence of the customers is relevant. Therefore, a brief summary of the pertinent literature in stochastic scheduling is appropriate.

Sequencing problems may be classified by a number of characteristics. The general form of the problem at hand would be defined by a single machine with N jobs to be processed, each with a release date (arrival time) to be determined. The

objective function in a scheduling problem typically is described in terms of release dates, r_j; completion times, D_j; due dates, d_j; and tardiness, $T_j = \max(0, D_j - d_j)$. The proposed objective function may be put in a stochastic scheduling context by defining $d_j = r_{j+1} = \tau_{j+1}$, in which case $W_j = T_{j-1}$, where W_j is the expected waiting time of the j^{th} customer.

These characteristics define a $1|r_j|\sum c_i T_i$ system [130], but with three unique characteristics. First and most importantly, the release dates are dependent on the sequence. That is, when the sequence is changed, the optimal arrival times under that sequence are generally altered. This is a situation not addressed in either deterministic or stochastic scheduling literature, with the exception of Wang's and Weiss's articles, as discussed below [162, 164]. Second, the due dates are dependent on the release dates. (This is not true of the deterministic analogue discussed in Appendix A, in which all due dates are fixed at τ_h.) Third, in the cost formulation discussed in Chapter III, the $(N+1)^{st}$ release date, τ_v, is fixed. The goal is to determine both the release dates and order of jobs that will minimize the objective function, under these special conditions.

If customers in the appointment system have identical service distributions and interarrival times are fixed and equal, so that customers differ only in cost functions, expected waiting times increase with the customer index. The optimal static strategy in this situation clearly is to sequence the job release dates in the order of decreasing unit costs. This is a special case of a strategy commonly called weighted shortest expected processing time first (WSEPT), in which expected processing times are weighted by the (linear) cost coefficients. The problem with generalizing this result is that, for the sequencing of arrivals to an appointment system, the scheduled arrival times shift as the sequence is altered, making it difficult to compare any improvement in optimal cost as the sequence of arrivals changes.

More difficult are cases for which neither cost functions nor service distributions are identical. Cox and Smith considered an M/G/1 queue in which customer classes

had different arrival rates, service distributions, and cost coefficients in a linear cost function. For the case in which preemption is forbidden, they proved the optimal dynamic strategy is WSEPT [26]. Kakalik and Little extended the work of Cox and Smith to allow the server to remain idle rather than serve a customer. They proved that WSEPT is still optimal and that the server should never choose to remain idle [78]. (This strategy of avoiding idle time is not optimal in the case of multiple servers, however [14].) This dependence of the optimal sequence only on the first moment of the service distribution is not likely to be the case for finite queues.

Another starting point is the literature regarding the sequencing of jobs that have identical release dates, random services, and in which the objective is to minimize the cost of tardiness. Rothkopf showed that in such cases, and for a large class of service distributions, allowing preemption will not improve the optimal cost [135]. Sevcik addressed the case of dynamically sequencing jobs with cost a linear function of tardinesses, identical release dates, idd general service distributions, and preemption allowed, in the context of a computer job queue. He showed the non-optimality of a WSEPT strategy and proved the optimality of a rule he called smallest rank (SR), in which remaining service times are weighted inversely by both the cost coefficient and by the probability the request will be processed within a sufficiently small time interval, and then ordered accordingly [142].

Glazebrook obtained the static and dynamic policies for a similar problem with stochastic services, identical release dates, an arbitrary set of precedence relations, and a cost that is a general function of the decision taken at each time. In rough terms, he proved that if the mean residual life of each service distribution is non-increasing, and if the unit cost of completing each individual job does not increase over the processing of that job, then an optimal strategy that optimally orders the customers at the start of the process does exist. This holds true whether preemption is allowed or not. He proposed a modification to Glazebrook's and Gitten's algorithm [50] to determine the optimal ordering [48].

The above approaches may have limited applicability, since the jobs have identical release dates. Pinedo considered optimal order when cost is defined as a weighted sum of job completion times, release dates are distinct, and preemption is allowed $(1|\text{pmtn}, r_j|\sum c_j D_j)$. He found that, if services are idd exponential, the optimal static and dynamic policies are WSEPT [129].

As mentioned in the heuristic appointment scheduling section, Weiss independently considered the static problem with release dates, distinct general services, and a cost that is a function of waiting and idle times, in the context of scheduling and sequencing surgeons to an operating room. He proved sequencing rules for the two-customer case but was unable to generalize them to larger numbers of customers. He proved that if the service distributions were exponential or Gaussian, the optimal sequence for two customers would have the customer with lowest variance service arrive first, but he was unable to extend this result to other service distributions or to larger numbers of arrivals. He simulated several sequencing examples to demonstrate the apparent efficacy of a smallest-variance-first scheme. This is the first known attempt to solve both the scheduling and sequencing problems [164].

Wang recently extended his earlier scheduling efforts to include the sequencing of multiple classes of customers. Customer services are assumed to be exponential but may have different means, and cost is a linear function of total waiting time and server availability (*i.e.*, $c_1 = \cdots = c_N$ and $\tau_v = 0$). He hypothesized that the optimal sequence orders arrivals by decreasing exponential rate and pointed out that this policy orders arrivals by increasing service variance. While there is empirical evidence for this conjecture when the overtime point is at zero, he was unable to prove it [162].

2.5 Optimization of Submodular Functions

The cost function to be discussed is both convex and submodular with respect to the arrival time vector, as long as the sequence of arrivals is fixed. A function

$f : \Re^n \rightarrow \Re$ is submodular on $\Re^n$ if

$$f(x \wedge y) + f(x \vee y) \leq f(x) + f(y) \quad \forall x, y$$

where $\wedge$ and $\vee$ are the meet and join operations, defined (for this purpose) by

$$\begin{aligned} x \wedge y &= [\min(x_1, y_1), \min(x_2, y_2), \ldots] \\ x \vee y &= [\max(x_1, y_1), \max(x_2, y_2), \ldots] \end{aligned}$$

Submodular functions were first explored by Lorentz [101]. Fan named them subadditive [37], while Marshall and Olkin coined the term L-subadditive, to avoid confusion with functions for which $f(x + y) \leq f(x) + f(y)$ [103]. However, submodular is the term in common use currently [157].

The problem of ordering the permutations of a vector relative to some submodular function has been addressed by a number of authors, and a recent survey can be found in Chang and Yao [20]. However, the cost function used in this dissertation is not submodular when the sequence of arrivals is altered, so these efforts are not relevant to the sequencing problem.

The problem of maximizing a submodular function has extensive application, and surveys can be found in two recent articles [47, 94]. Although the work is equally applicable to minimization of a supermodular function, it is not of help in minimizing a submodular function, so it is not reviewed here.

Topkis proved that the set of points over which a submodular function attains its minimum is a sublattice. (M is a sublattice of L if $M \subseteq L$ and $x, y \in M$ implies $x \wedge y \in M$ and $x \vee y \in M$.) Using this result, he proposed a general approach to minimization [155, 156]. He pointed out a number of problems that hinge on modification of a submodular function, including: the min-cut, max-flow problem in graph theory; an optimal pricing strategy problem; and optimal control

of an unreliable system. The approach advocated in this dissertation is in a sense a modification of his approach.

Goemans and Ramakrishnan recently addressed the problem of finding the minimum cut in a graph and also framed it in terms of minimizing a submodular function [52]. This effort was apparently independent of Topkis's.

2.6 Summary

The major goal of this research is to determine the optimal sequence and schedule of customer arrivals efficiently, given various circumstances. The approach to the scheduling problem will be analytical, while the sequencing problem will be treated heuristically. A summary of the relevance of the above literature to these goals follows.

The research in the area of control of queues appears peripheral to this effort, although perhaps another researcher might formulate the problem differently and find an application of these approaches to the problem. The heuristic appointment literature holds some promise in guiding this research toward realistic problems, although the modeling and empirical approaches used in their schedule optimizations are not applicable here.

This dissertation would be classified as part of the theoretical appointment literature, and it is therefore not surprising that most of the efforts in that section are relevant here. Liao's approach to lattice schedule optimization [95, 96, 97] was successful and must be considered as an alternative to the approach that will be proposed here. The various NLP approaches to schedule optimization must also be considered [60, 61, 126, 160]. The approach to cost evaluation that will be proposed here is similar to Wang's embedded continuous Markov chain approach [57, 160, 163]. The schedule optimization approaches that will be advocated for various circumstances are refinements of Simeoni's basic idea [145, 158], which bear a close relationship

to Topkis's framework [155, 156]. The other theoretical appointment literature discussed is also relevant, albeit less directly so.

Very few efforts have addressed appointment sequencing. Wang's unpublished attempt at sequence optimization for exponential services - albeit imaginative – was unsuccessful and quite limited in scope [162]. Likewise, Weiss obtained the optimal sequence for systems with only two customers, limiting its usefulness [164]. The methodologies used in these efforts yield little promise for more general problems. Based on results in Appendix A for optimally sequencing deterministic appointment systems, analytical approaches will be rejected in favor of a good heuristic. No previous research has addressed the sequencing of customers with deterministic services to an appointment system.

III. Objective Function Formulation

In many optimization schemes, evaluation of the objective function consumes the majority of the computation involved. It is therefore important to examine methods of efficiently evaluating the cost function. This chapter examines cost evaluation under a variety of circumstances.

Recall that τ_i is the scheduled arrival time of customer i and is controllable. The waiting time of the i^{th} customer for a particular realization is W_i, often expressed in this chapter as $W_i(\tau)$ or $W_I(\chi)$ to emphasize the dependence of the waiting time on the vector of arrival times or service times. The unit cost of the i^{th} customer's waiting time is c_i. Customers are indexed in order of their arrivals.

The cost is considered to be a convex combination of expected waiting times for the N customers and the expected server overtime, where server overtime is the time past τ_v (overtime point) that the server must continue to serve customers. The coefficients in this convex combination are the c_i. The usual requirement for convex combinations that $\sum_{i=1}^{N} c_i = 1.0$ will sometimes be useful, but will usually be ignored, since the problem can always be transformed to suit this requirement by scaling the unit costs, as long as they are nonnegative and at least one is positive. This condition will always be assumed. The unit costs are also assumed to be constant.

For the purposes of calculation and notation, it is convenient to add a fictitious $(N+1)^{st}$ customer to the schedule at τ_v. This new customer is not permitted service until the N^{th} customer completes its service. Thus, the expected server overtime is $E[W_{N+1}(\tau)]$.

Given these definitions, the total cost associated with a particular arrival time vector is

$$C(\tau) = \sum_{i=2}^{N+1} c_i E[W_i(\tau)] \tag{1}$$

This cost is thus a function of the arrival time vector τ, the overtime point, τ_v, and the expected waiting time vector $E[W(\tau)]$, which in turn is a function of τ, the service probability density functions (PDFs), $f_1, f_2, \ldots f_N$, and the probabilities of customers arriving for appointments, $\gamma_1, \gamma_2, \ldots, \gamma_N$. The first arrival time is fixed at 0, and the constraint $\tau_i \geq 0$ applies for all i. The constraints $\tau_i \leq \tau_h \ \forall i$ apply if arrival times are bounded, where τ_h is called the schedule horizon. The constraints $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_N$ apply if the sequence of arrivals is to remain fixed. The task at hand is to efficiently determine or approximate the expected waiting time for each customer, given general service distributions and prompt arrivals unless failing to show.

Two approaches are examined here. The first is to obtain an exact expression for expected waiting times for general service distributions. The expression for the j^{th} customer turns out to be a j-fold convolution integral, so it generally will be necessary to approximate. The second approach is to approximate the service distributions with phase-type distributions or with Erlang distributions and exploit the memoryless property of the phases. Also, the simplifications that accrue from restricting arrival times to lattice points are examined. Last, a brief look at some properties of the cost function provides the reader with a better understanding of the optimization problems treated in subsequent chapters.

3.1 General Service Distribution

Assume for the moment that no-shows are forbidden (*i.e.*, $\gamma_k = 1 \ \forall k$). The expected waiting times can be determined by a convolution argument. For each possible vector of service times χ, it is well-known [46] that

$$W_j = \text{MAX}\left[0, W_{j-1} + \chi_{j-1} - \tau_j + \tau_{j-1}\right] \tag{2}$$

Then for $z \geq 0$,

$$\begin{aligned} P(W_j \leq z) &= P(W_{j-1} + \chi_{j-1} - \tau_j + \tau_{j-1} \leq z) \\ &= P(W_{j-1} + \chi_{j-1} \leq \varphi) \\ &= \int_0^{\varphi} P(W_{j-1} \leq \varphi - \chi_{j-1})\, dF_{j-1}(\chi_{j-1}) \end{aligned} \tag{3}$$

where $F_j(\chi)$ is the cumulative distribution function (CDF) associated with $f_j(\chi)$, and $\varphi = \tau_j - \tau_{j-1} + z$. This expression is just the transient form of Lindley's waiting time result [99]. From this equation, the waiting time PDFs can be obtained recursively. To obtain the expected waiting times, recall that $E(W_j) = \int_0^{\infty} P(W_j \geq z)\, dz$. Then

$$E(W_j) = \int_0^{\infty} \left(1 - \int_0^{\varphi} P(W_{j-1} \leq \varphi - \chi_{j-1})\, dF_{j-1}(\chi_{j-1})\right) dz. \tag{4}$$

Now consider the more general case where customers may fail to show but are otherwise punctual. Define a show vector, θ, where $\theta_j = 1$ if customer j showed for the appointment in a particular instance and $\theta_j = 0$ otherwise. The plan is to determine $E(W_j \mid \theta_j = 1)$ by conditioning on the value of θ_{j-1}. With probability $1 - \gamma_{j-1}$, the service time of customer $j-1$ is zero, and its waiting time is also zero. However, just for the purposes of calculating $E(W_j \mid \theta_j = 1)$, one can picture that customer $j-1$ always showed, waited for service, and then either completed service immediately, with probability $1 - \gamma_{j-1}$, or else underwent service of length χ_{j-1}, with probability γ_{j-1}. Since $E(W_1) = 0$, for $j \geq 2$,

$$\begin{aligned} E(W_j \mid \theta_j = 1) &= (1 - \gamma_{j-1}) \int_0^{\infty} P(W_{j-1} > \varphi \mid \theta_{j-1} = 1)\, dz \\ +\gamma_{j-1} \int_0^{\infty} \int_0^{\varphi} &(1 - P(W_{j-1} \leq \varphi - \chi_{j-1} \mid \theta_{j-1} = 1))\, dF_{j-1}(\chi_{j-1}) dz \end{aligned} \tag{5}$$

Since $E(W_j \mid \theta_j = 0) = 0$, it is a simple matter to obtain customer j's expected waiting time by conditioning upon whether it arrives or not:

$$E(W_j) = \gamma_j E(W_j \mid \theta_j = 1) \tag{6}$$

In general, these integrals will not have an analytic solution and will require numerical approximation. Since multiple convoluted integrals must be evaluated, the potential for approximation errors is compounded. This approach clearly is computationally oppressive for most service distributions and for problems of realistic size.

3.2 Coxian Distribution

A phase-type (PH) distribution is defined as the distribution of times required to transit a network of exponential stages, or phases. The Coxian-r distribution is a phase-type distribution defined here by the particular network in Figure 3. In this figure, $b_1, b_2, \ldots b_r$ represent routing probabilities as shown and $\mu_1, \ldots \mu_r$ represent the exponential phase rates at each stage. Unlike Cox's original formulation, the routing probabilities will be constrained to be positive, and the phase rates will be positive and real [25]. This is still sufficiently general to model distributions with support on $\Re^+$ (the set of positive real numbers).

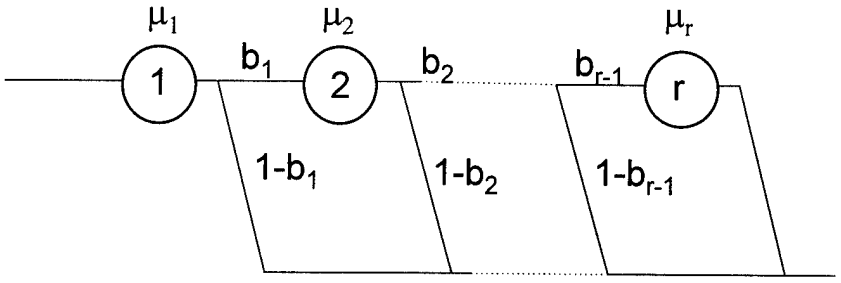


Figure 3. Series-parallel stages representing a Coxian-r distribution

The utility of the Coxian is twofold. First, with appropriate choice of number of stages, service means and routing probabilities, it can approximate most distributions. Cox demonstrated it can represent exactly any distribution whose PDF has a rational Laplace transform, if complex transition rates are allowed [25], and Newman and Reddy showed that the Laplace transform of any PDF may be approximated arbitrarily closely by a rational function [119]. Thus, a Coxian distribution can be used to approximate any general distribution [3]. If the support for a PDF is $\Re^+$, a Coxian distribution with real transition rates and positive routing probabilities suffices to approximate the PDF to any desired accuracy [46]. Approaches to parsimoniously approximating a distribution with a Coxian distribution are discussed in Appendix F.

Second, since the Coxian is a sum of fractions of convolutions of exponentials, Coxian service distributions yield Markovian stochastic processes that, due to the memoryless property of the exponential distribution, simplify the analysis of many systems in fields such as queueing theory, insurance risk theory, renewal theory, and reliability [6].

Consider Figure 3 as the state transition diagram depicting the service distribution of a single customer. No-shows are allowed, but will be modeled externally to the distribution. (While the show rate could be considered by adding a routing around the first phase, and one could thus consider $\gamma = b_0$, this would prevent a phase-type representation, essential to the argument to follow.) Suppose that a Coxian process is in stage k at time t_0. Let $p_i(t)$ be the row vector representing the probability that the process is in the i^{th} stage (state) at time t. If $i < k$, define $p_i(t) = 0$. Then the differential equations describing this pure birth process for a single customer are:

$$\frac{dp_k(t)}{dt} = -\mu_k p_k(t)$$

$$\frac{dp_i(t)}{dt} = -\mu_i p_i(t) + b_{i-1}\mu_{i-1}p_{i-1}(t) \ \forall i : k < i$$

with initial conditions $p_k(t_0) = 1$ and $p_i(t_0) = 0\ \forall i : i \neq k$. Here, the exit state is defined as the $(r+1)^{st}$ state. Define $b_r = 1$ for consistency. The probability of exiting by t is just $1 - \sum_{i=k}^{r} p_i(t)$.

If $\mu_1 = \ldots = \mu_r = \mu$ and $b_1 = \ldots = b_{r-1} = 1$, then the sojourn time is Erlang-$r(\mu)$ distributed, and $p_i(t)$ follows a truncated Poisson distribution. In other cases, the solution of these differential equations does not lead to a convenient form and, for all but the smallest cases, is burdensome to calculate by hand. For this reason, such cases are often handled by constructing the infinitesimal transition matrix Q. The solution is then

$$p(t) = p(t_0)e^{\left(\int_{t_0}^{t} Q dt\right)} = p(t_0)e^{[Q(t-t_0)]} \tag{7}$$

The transition matrix without absorbing state and for a single customer is

$$T = \begin{bmatrix} -\mu_1 & b_1\mu_1 & 0 & 0 & \cdots & 0 \\ 0 & -\mu_2 & b_2\mu_2 & 0 & \cdots & 0 \\ 0 & 0 & -\mu_3 & b_3\mu_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & -\mu_{r-1} & b_{r-2}\mu_{r-2} & 0 \\ 0 & 0 & \cdots & 0 & -\mu_{r-1} & b_{r-1}\mu_{r-1} \\ 0 & 0 & \cdots & 0 & 0 & -\mu_r \end{bmatrix}$$

To include the exit state, define $T_0' = \begin{bmatrix} (1-b_1)\mu_1 & (1-b_2)\mu_2 & \cdots & \mu_r \end{bmatrix}^T$, which represents the transition probabilities to the exit state. Then

$$Q = \left[\begin{array}{c|c} T & T_0' \\ \hline 0 & 0 \end{array}\right] \tag{8}$$

Substitution into Equation (7) yields the probability vector of the number of stages completed or bypassed at time t, assuming that $p(t_0)$ is known, that no customers

are scheduled to arrive between t_0 and t, and that only one customer is waiting for service at t_0. The task now is to build an algorithm that eliminates these assumptions and produces the desired probabilities of being in each state at each time.

3.3 Cost Evaluation Algorithm for Coxian Service

The distribution of completion time of multiple customers may be found by expanding the matrix. Assume for now that all customers are initially available for service. Let $T_{i,1}$ and $T'_{i,0}$ be the transition matrix and the exit probability vector for the i^{th} customer, of size $r_i \times r_i$ and r_i, respectively. Construct $T_{i,0}$ by appending $r_{i+1} - 1$ columns of zeros to $T'_{i,0}$. Build Q as above for the first customer, but add r_i states for each subsequent customer, using the first state of each subsequent customer as the exit state of the previous customer. Thus, if customer i is in its j^{th} stage of service, the system is in state $j + \sum_{k=1}^{i-1} r_k$. Now Q may be defined as the block bidiagonal form

$$Q = \begin{bmatrix} T_{1,1} & T_{1,0} & 0 & 0 & \cdots & 0 & 0 \\ 0 & T_{2,1} & T_{2,0} & 0 & \cdots & 0 & 0 \\ 0 & 0 & T_{3,1} & T_{3,0} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & T_{r,1} & T'_{r,0} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

If there is a possibility of a customer failing to enter the system, subsequent to the currently served customer, that possibility can be adjusted for in the following manner. Let $j = 1$. Consider the exit phase column of T for customer j, which is the initial phase of $j + 1$. The probability of customer $j + 1$ showing is γ_{j+1}, so move $1 - \gamma_{j+1}$ of each entry in j's exit phase column to the corresponding entry in the exit column for $j + 1$. This is the new T. Increment j and repeat if $j \leq N + 1$.

The resulting transition matrix accounts for all no-shows but the first, given that all customers either arrive at $t = 0$ or fail to show.

In the cases of arrivals at other times than $t = 0$, Equation (7) fails; by the nature of its exponential phases with real transition rates, it can only represent distributions with support on $[0, \infty)$, and an arrival at $t \neq 0$ would require support on $[t, \infty)$. A piecewise strategy overcomes this problem. Assume $p(\tau_j)$ is known. Define a transition matrix Q_j that accounts only for those customers currently present in the system, and apply Equation (7) only up to the time of the next arrival to obtain $p(\tau_{j+1})$. Then add in the r_{j+1} stages of customer $j + 1$ to get Q_{j+1}. Repeat as necessary to find the probability vector at the desired time.

The possibility of an initial no-show is a problem in the application of Equation (7) as well; it would imply the existence of a phase with instantaneous service, which can only be represented in Q by the limit as the initial phase rate of customer $(j+1)$ goes to infinity (a Bernoulli distribution). However, with the piecewise strategy, this problem is avoided. At each arrival epoch, one need only modify $p(\tau_j)$ by shifting a fraction $(1-\gamma_j)$ of the probability in j's arrival state to its exit state before applying Equation (7).

In practice, Q_j need never be formed. Instead, Q itself can be used. Although it generates $p(\tau_{j+1})$ from $p(\tau_j)$, which may have a nonzero probability of being in infeasible states (*i.e.*, states past $\sum_{i=1}^{j} r_i + 1$), this infeasible probability mass is merely the probability that further service would have taken place or bypassed had other customers been available. As such, it can be accounted for by summing the probabilities of being in infeasible states and transferring this probability mass to the exit state of customer j. Next, $1 - \gamma_{j+1}$ of the mass in customer j's exit state should be transferred to customer $(j + 1)$'s exit state, to account for the possibility of customer $(j + 1)$ failing to show. This simplifies the calculation of $p(t)$.

Now that the probability of being in each state at each time is determined, a last construct is required to determine expected waiting times. Define a conditional

expected waiting time vector $\Omega(j)$, in which $\Omega(j)_i$ represents the expected waiting time of customer j conditioned on the state upon j's arrival being i. (It is defined as a set of vectors, rather than an array, only for notational convenience.) Let $X_{h,i}$ be the expected remaining service time of customer h, given that it is in the i^{th} stage of its service. Let $b_{h,i}$ be the transition probability from the i^{th} phase to the $(i+1)^{st}$ phase of the h^{th} arrival, and let $\mu_{h,i}$ be the transition rate of the i^{th} phase of the h^{th} arrival. Then by inspection of Figure 3,

$$X_{h,i} = \begin{cases} \mu_{h,i}^{-1} & i = r_h \\ \mu_{h,i}^{-1} + b_{h,i} X_{h,i+1} & 1 \leq i < r_h \end{cases} \tag{10}$$

can be used to recursively obtain all of X. If the current customer is h and it is in its i^{th} stage of service, then by inspection of Figure 4,

$$\Omega(j)_h = \begin{cases} 0 & j \leq h \\ X_{h,i} & j = h+1 \\ \Omega(j-1)_h + \gamma_{j-1} X_{j-1,1} & j > h+1 \end{cases} \tag{11}$$

can be used to obtain all values of each vector $\Omega(j)$. The expected waiting time for customer j, assuming it does not have to wait if it does not show, is then

$$E(W_j) = \gamma_j \Omega(j) p(\tau_j). \tag{12}$$

Wang and Gray accomplished the same result for the case of iid Erlang services by a scheme that requires the inversion of T [57, 163]. Equations (10) and (12) provide an efficient alternative to their scheme, accounting for idd services and avoiding the potential floating point problems discussed in Appendix G.

The following algorithm summarizes the procedure outlined above for obtaining the expected waiting times for Coxian service with no-shows.

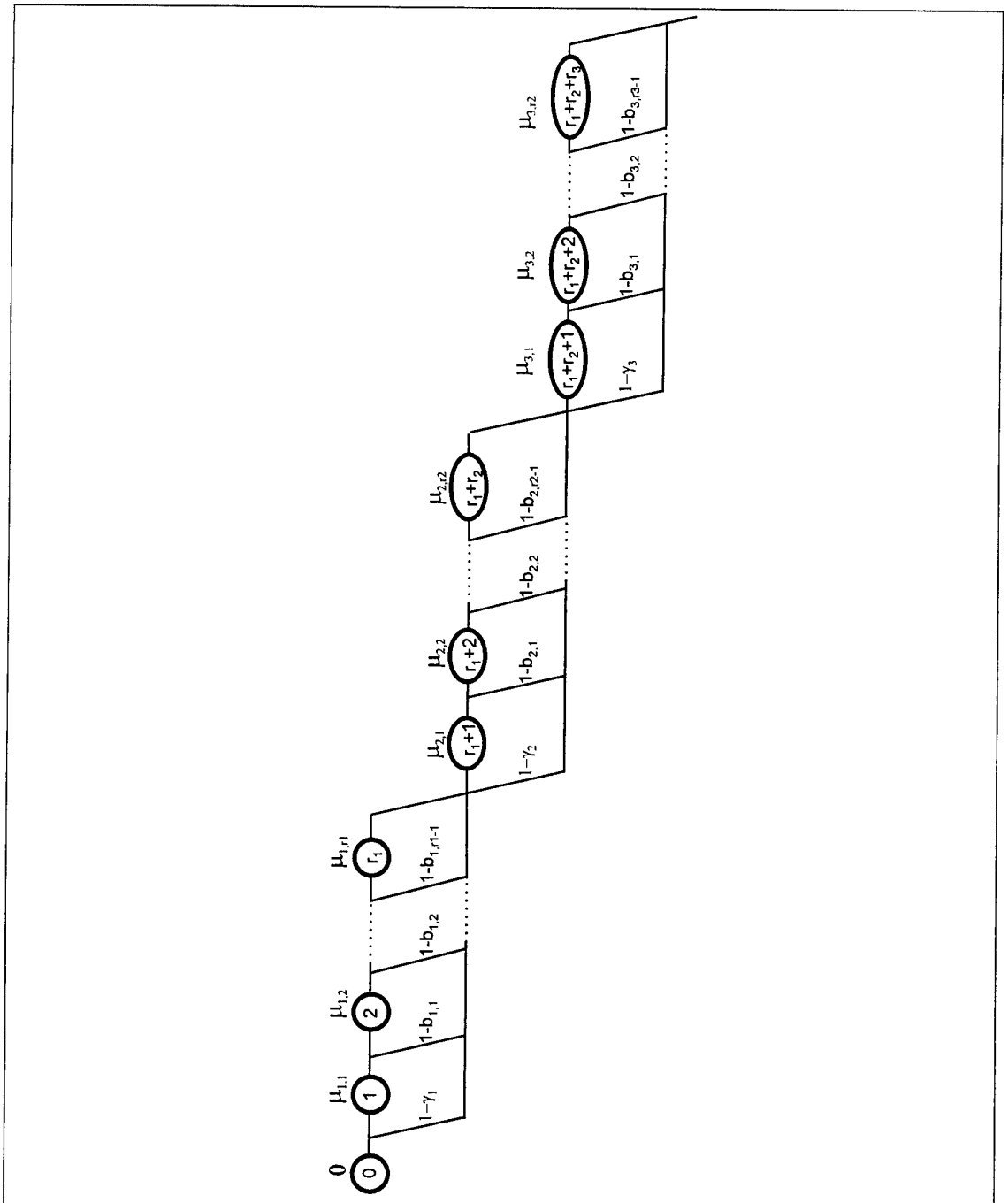


Figure 4. Coxian series. State diagram representing a series of three customers with idd Coxian-r_i services and show rates

Cost Evaluation Algorithm For Coxian Service

1. Set $E(W_1) = 0$. Set $p(0) = [\ 1 \quad 0 \quad 0 \quad \ldots \quad 0\ ]$. Set $j = 1$. Set $h = r_1$
2. Account for no-shows by replacing $p(\tau_j)[h+1]$ with $\gamma_j p(\tau_j)[h+1]$ and $p(\tau_j)[h+r_{j+1}+1]$ with $(1-\gamma_j)p(\tau_j)[h+1]$.
3. Let $p(\tau_{j+1}) = p(\tau_j)\exp[Q(\tau_{j+1}-\tau_j)]$. Let $h = h + r_j$.
4. Let $p(\tau_{j+1})[h+1] = \sum_{i=h+1}^{N+1} p(\tau_{j+1})[i]$. Set $p(\tau_{j+1})[i] = 0$ for each $i > h+1$. If $j < N$, let $j = j+1$ and return to step 2.
5. Obtain the expected waiting time vector from Equations (10), (11), and (12).
6. Apply Equation (1) to obtain the cost associated with this schedule.

This algorithm works for either lattice or continuous arrival times. In the case of lattice arrival times, some simplifications may be made, and these are discussed in Section 3.5.

When arrivals are not restricted to lattice times, there is not a practical reason to consider the waiting time in the event two customers are scheduled to arrive at the same time; such a schedule cannot be optimal unless the cost of customers waiting is zero. However, the algorithm does work for simultaneous arrivals. More efficient approaches to simultaneous arrivals will be addressed in detail in Section 3.5, which discusses lattice arrival times.

The accuracy of the above cost evaluation is mainly dependent on the accuracy of the matrix exponentiation in step 3, since the remaining of the operations involve only addition and subtraction. The accuracy of the matrix exponentiation used is discussed in detail in Appendix G. That section concludes that exponentiation becomes less accurate as two phase rates become extremely close without coinciding, or as one of the phase rates diverges from the others. If neither of these situations obtains, the exponentiation is assumed here to be accurate.

3.4 Erlang Service Distribution

The above formulation is feasible but is computationally intensive if many cost evaluations are to be made. If the service distributions are iid, have a coefficient of variation less than or equal to one, and it is acceptable to approximate the second moment rather than match it exactly, an Erlang approximation leads to further savings in computation. The Erlang distribution with r stages has PDF

$$f(t) = \frac{\mu e^{-\mu t}(\mu t)^{r-1}}{(r-1)!}, \quad t \geq 0 \tag{13}$$

To approximate a given PDF, the parameter μ is chosen to be the first moment of the distribution. Given variance (or sample variance, if one is approximating an empirical PDF) of σ^2, the number of stages is selected by

$$r = \frac{1}{\text{int}\,[\sigma^2 \mu^2]} \tag{14}$$

Since the variance will be approximated by the inverse of an integer, the approximation will be more accurate for smaller variances.

Liao first obtained the expected waiting times for the case of iid lattice arrival times with an Erlang service distribution [95, 96, 97]. Here, the case of arrival times not restricted to a lattice is examined first. Since the Erlang distribution with r stages is just a special case of the Coxian with r iid stages with mean service rate μ and transition probabilities all set to 1.0, the above algorithm for Coxian services could be used to obtain expected waiting times. Computation in this case is speeded greatly, since

$$Q = \begin{bmatrix} -\mu & \mu & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\mu & \mu & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\mu & \mu \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{15}$$

and $e^{Q(t-t_0)}$ becomes an upper triangular matrix for which the (i,j) entry follows a truncated Poisson distribution:

$$e^{Q(t-t_0)}(i,j) = \begin{cases} 0 & i > j \\ \frac{e^{-r\mu(t-t_0)}(r\mu(t-t_0))^{j-i}}{(j-i)!} & i \leq j < rN+1 \\ 1 - \sum_{m=0}^{j-1} \frac{e^{-r\mu(t-t_0)}(r\mu(t-t_0))^{m-i}}{(m-i)!} & j = rN+1 \end{cases} \tag{16}$$

For the last column (index $j = rN+1$), the entries are sums of all the Poisson probabilities for $j \geq rN+1$. All entries are easily calculable with a recursive routine, reducing computation when calculating $e^{Q(t-t_0)}$. No further simplifications are helpful for the case of Erlang-r services with unrestricted arrival times.

3.5 Lattice Arrival Times

For lattice arrival times, simplifications arise from two sources. Let Δ be the smallest allowable time interval. First, $e^{Q\Delta}$ may be calculated at the beginning of the algorithm, obviating the need to calculate $e^{Q(t-t_0)}$ at any iteration, since $t - t_0 = k\Delta$ for some integer k, and $e^{Qk\Delta} = \left(e^{Q\Delta}\right)^k$. This substantial simplification can also be applied to continuous cases by imposing a lattice that allows approximation of the arrival times; that is, by setting Δ equal to the largest number that is (approximately) an integral factor of each of the interarrival times. It should be noted that, if the overtime point is later than the last arrival time and is not lattice, another exponentiation must be performed. The computer program EVALUATE in Section H.1 approximates the overtime point by the nearest lattice point in order to avoid this second exponentiation.

Second, the likelihood of multiple arrivals at an instant is no longer infinitesimal. While the Coxian-r algorithm can be used, there is a faster approach. Let k be the index of the first customer scheduled to arrive at τ_k, and let $v(\tau_k)$ be the number of customers scheduled to arrive at τ_k. $E(W_k)$ may be found as usual. For subsequent customers, $E(W_{k+i}) = E(W_{k+i-1}) + \frac{\gamma_{k+i}r}{\mu}$, for $i \in [1, v(\tau_k) - 1]$. In the

special case of lattice arrival times, iid Erlang-r services, and $\gamma_1 = \cdots = \gamma_N = \gamma$, $E(W_{k+i}) = E(W_k) + \frac{i\gamma r}{\mu}$, and the total expected waiting time of customers arriving at t_k is

$$\sum_{i=0}^{v(\tau_k)-1} E(W_{k+i}) = v(\tau_k)E(W_k) + \frac{\gamma r v(\tau_k)(v(\tau_k)-1)}{2\mu} \tag{17}$$

The following summarizes the cost algorithm in the case of iid Erlang service, lattice arrival times, and constant show rate.

Cost Evaluation Algorithm For iid Erlang Service

1. Construct $e^{Q\Delta}$ by using Equation (16). Let $p(0)_i = [\ 1 \quad 0 \quad \ldots \quad 0\ ]$. Let $j = 1$. Let $\tau_0 = 0$.
2. Let $v(\tau_j)$ be the number of arrivals at τ_j. Let $k = \left(\tau_j - \tau_{j-1}\right)/\Delta$.
3. Let $p(\tau_j) = p(\tau_{j-1})(\exp[Q\Delta])^k$.
4. Let $p(\tau_j)_{rj+1} = \sum_{i=rj+1}^{rN} p(\tau_j)_i$. Let $p(\tau_j)_i = 0$ for $i > rj+1$. This shifts all the infeasible probability mass to the exit state of j. Let $h = 0$.
5. Let $p(\tau_j)_{r(j+h)+1} = \gamma p(\tau_j)_{r(j+h)+1}$. Let $p(\tau_j)_{r(j+h+1)+1} = (1-\gamma)p(\tau_j)_{r(j+h)+1}$. If $h < v(\tau_j)$, then increment h and repeat this step. This adjusts $p(\tau_j)$ for no-shows of arrivals at τ_j.
6. Find the sum of expected waiting times for arrivals j through $j + v(\tau_j) - 1$ using Equations (2) and (17).
7. Let $j = j + v(\tau_j)$. If $j < N+1$, return to step 2.
8. Apply Equation (1) to obtain the cost associated with this schedule.

3.6 Modeling Lateness

The previous cost formulation assumed punctual customers if the customer joined the queue at all, but allowed for the possibility of no-shows. This is the case

addressed in the majority of this dissertation because of the number of problems faced in optimizing appointment systems where lateness is permitted. It is, however, possible to obtain the cost of an appointment system in which each customer has some lateness distribution, and that is shown in this section. The basic approach is to roughly model the lateness distribution as a bounded discrete distribution and prorate the cost of each possible resulting realization of the lateness values by its probability of occurrence.

Several authors have examined the effects of lateness on an appointment system [137, 168], but the continuous distribution approaches used would preclude the imbedded Markov approaches used so far in cost evaluation, so these earlier approaches are rejected. Here, the j^{th} customer is assumed to have a lateness distribution $\ell_j(t)$ that is idd, discrete with support at lattice points only, bounded below by 0, and bounded above by $\tau_h - \tau_j$. The lower bound of 0 is for convenience only and does not preclude a customer arriving early; in this case, one need merely redefine its arrival time to the earliest possible and adjust the lateness distribution accordingly.

Suppose the lateness probabilities for customer j are nonzero for ε_j values, all of which are schedule lattice points. Because the lateness distributions are independent, the marginal probability of a given realization of latenesses is just the product of the probabilities of each lateness occurring. This leads to the formulation of the system as a set of $\prod_{j=1}^{N} \varepsilon_j$ instances (in which the customers are each punctual). The waiting time of customer j is just the sum of the product of the waiting times for j obtained in an instance and the marginal probability of that instance, over the possible instances.

Assuming that the customers are served in the order in which they were scheduled, rather than in the order they actually arrived, the same transition matrix Q can be used for each sub-problem as was used in the punctual case defined earlier. Because the state space is not increased, and since only a single matrix exponentiation need be performed still, the additional calculations required to assess cost when

lateness is allowed are not extensive. However, because the number of sub-problems that must be evaluated is dependent on $\prod_{j=1}^{N} \varepsilon_j$, the number of possible values of lateness to be considered for each customer will affect the calculation speed.

If the service discipline is FIFO, on the other hand, customer order may not be the same for each possible realization of the lateness. In such a case, Q must be reconstructed and re-exponentiated for each possible service order. This will lead to longer run times for the cost evaluation algorithm under FIFO.

It is seen that the cost of an appointment system in which customers can be late or early can be approximated quite effectively. However, this dissertation does not consider the effect of lateness further, since the optimization algorithms to be presented will depend on convexity of the cost function and the equivalence of the scheduled and the actual arrival order. While consideration of the effects of lateness on the optimal schedule policy is undoubtedly important, that effort must be relegated to future research.

3.7 The Nature of the Objective Function

With the above evaluation tools available, the nature of the cost function is more easily explored. Examination of some rough plots will help the reader understand the nature of the function and the task of finding the optimal schedule and sequence. In Figure 5, a fixed time horizon of $t \in [0, 30]$ was imposed, in which three identical customers with iid exponential service distributions must be scheduled. Costs are linear functions, and the unit costs c_1, c_2, and c_3 are equal. The unit overtime cost, c_4 in this case, is arbitrarily set equal to c_1 and the overtime point is set to τ_h. The probability of a no-show is set to zero. The first customer's arrival time is fixed at $\tau_1 = 0$, and the abscissae represent τ_2 and τ_3. Several insights can be gleaned. First, the plot must be symmetric about $\tau_2 = \tau_3$, since the customers are identical in every way. Even if the customers were not identical, the line $\tau_2 = \tau_3$ would divide the plot into two convex regions (as will be proved in Section 4.1).

This line represents a ridge; waiting time is locally maximal along coordinate axis directions when two customers are scheduled to arrive at the same time.

Minima seem to occur when the customers are scheduled at roughly equal time intervals. However, the interarrival times are not in general equal for the optimal schedule with identical customers, as the following argument shows. Consider a very simple case: two customers, with a fictitious third customer fixed at $\tau_3 = \tau_h$ to account for server overtime, and the first fixed at $\tau_1 = 0$. The wait imposed on customer 2 is a decreasing function of τ_2, and the wait imposed on customer 3 is an increasing function of τ_2. If the sum of the service times of customers 1 and 2 never exceeded τ_3, the first function would be a reflection of the second about $\tau_2 = \tau_3/2$, and the minimum of their sum is clearly attained at $\tau_2 = \tau_3/2$. When the possibility increases that the sum of the services of customers 1 and 2 exceeds τ_3, customer 3's wait also increases, but customer 2's wait is unaffected. This disturbance of the symmetry will shift the optimal value of τ_2 lower (as $P[\chi_1 + \chi_2 > \tau_3]$ increases).

Figure 6 depicts a case in which four customers are to be scheduled in the fixed horizon $t \in [0, 40]$. For the purposes of the plot, $\tau_1 = 0$ and $\tau_4 = 30$ are fixed, while τ_2 and τ_3 are plotted on the x and y axes, as before. Services are iid exponential with mean of 5 time units for customers 1, 2, and 3, while customer 4's service is exponential with mean of 10 time units. Costs are linear with equal coefficients, as before. Again, the plot is piecewise convex for a particular order of customers, reaching local maxima as two or more customers are scheduled to arrive at the same time (proved in Section 4.1). The task of locating the minimum could be considered twofold. One problem is to locate the optimum schedule for each order of customers (the scheduling problem), while the second is to find the smallest of these optima (the sequencing problem). Efficient pursuit of these two objectives to obtain the best appointment policy is the goal of the rest of this dissertation.

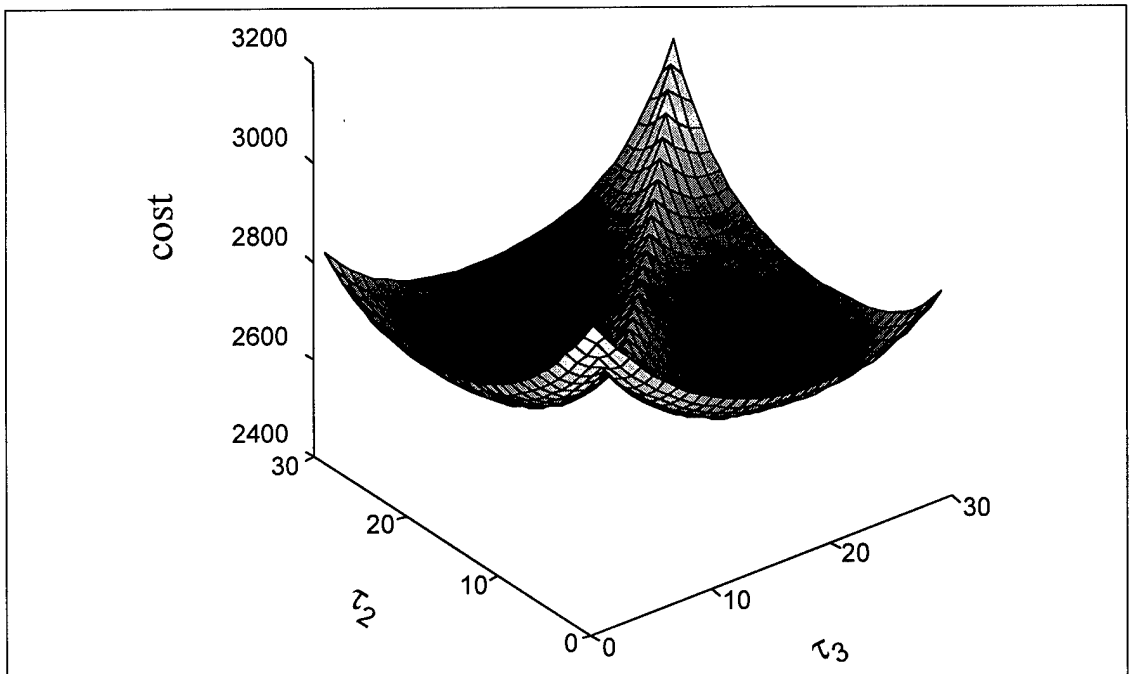


Figure 5. Cost function: Example 1. Plot of $C(\tau)$ vs τ_2 and τ_3 for a 3-customer example

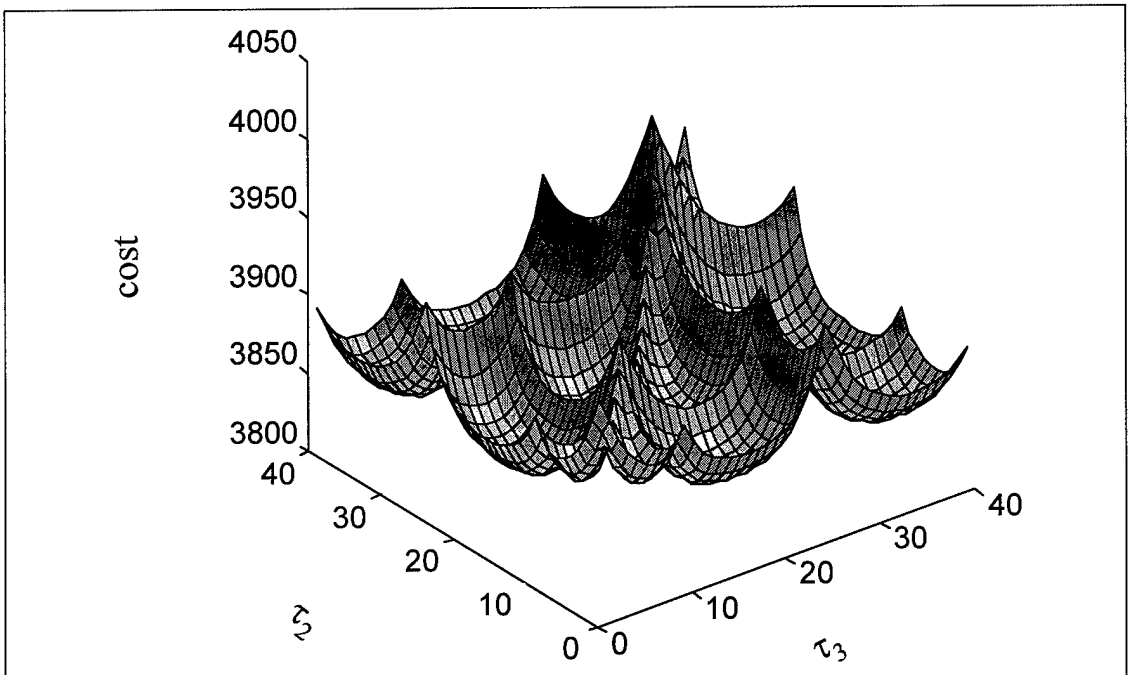


Figure 6. Cost function: Example 2. Plot of $C(\tau)$ vs τ_2 and τ_3 for a 4-customer example

IV. Scheduling Arrivals When the Sequence is Fixed

The goal of this chapter is to provide an effective tool to determine the optimal schedule of arrivals, given that the order of arrivals is fixed. This is tantamount to searching the arrival time space over the region for which the constraints $\tau_1 \leq \tau_2 \cdots \leq \tau_N$ hold.

It was claimed in Section 1.2 that one may set $\tau_1 = 0$ without losing generality, since an optimal schedule will always have the first customer arrive at the start of the server's availability. A later arrival time would incur increased server idle cost without a reduction in total waiting cost. An earlier arrival time would increase waiting time for the first customer while failing to reduce idle costs or other customers' waiting costs.

Since, as will be proven, the cost function is convex with respect to τ in the admissible region, an optimum is assured, and any number of nonlinear optimization schemes may be applied to obtain the continuous optimum. For instance, Healy *et al.* applied a Hooke-Jeeves optimization [60, 61], while Wang and Gray applied both a simple gradient search and a conjugate gradient search [57, 160].

For lattice arrival time problems, enumeration is not possible unless a finite time horizon is imposed, and even then it is prohibitive for moderately-sized problems. Suppose there are K time slots and N customers, the first of which has a fixed arrival time. By a simple combinatorial argument, there are $\binom{N+K-2}{N-1}$ possible schedules to evaluate. A problem involving 20 customers and 48 time slots would generate $\binom{66}{19} \approx 1.73 \cdot 10^{16}$ candidate schedules. Clearly, another approach is required.

Liao *et al.* utilized a bound provided by the optimal dynamic schedule and applied a branch-and-bound technique to solve such problems [95, 96, 97]. Simeoni applied a variation on a simple coordinate search that was based on the convexity and submodularity of the cost function [145]. The approach to be advocated here

for both lattice and continuous arrival times is a substantial extension of Simeoni's coordinate search.

An example may help focus on the pertinent aspects of this optimization for lattice arrival times. Figure 7 shows the cost of several schedules with 3 customers and 5 time slots. Unit waiting costs and overtime cost are identical ($c_2 = c_3 = c_4$), $\Delta = 1$, and the overtime point is located one slot past the schedule horizon ($\tau_v = \tau_h + 1 = 5$). Services are iid exponential distributions with a mean of 1.

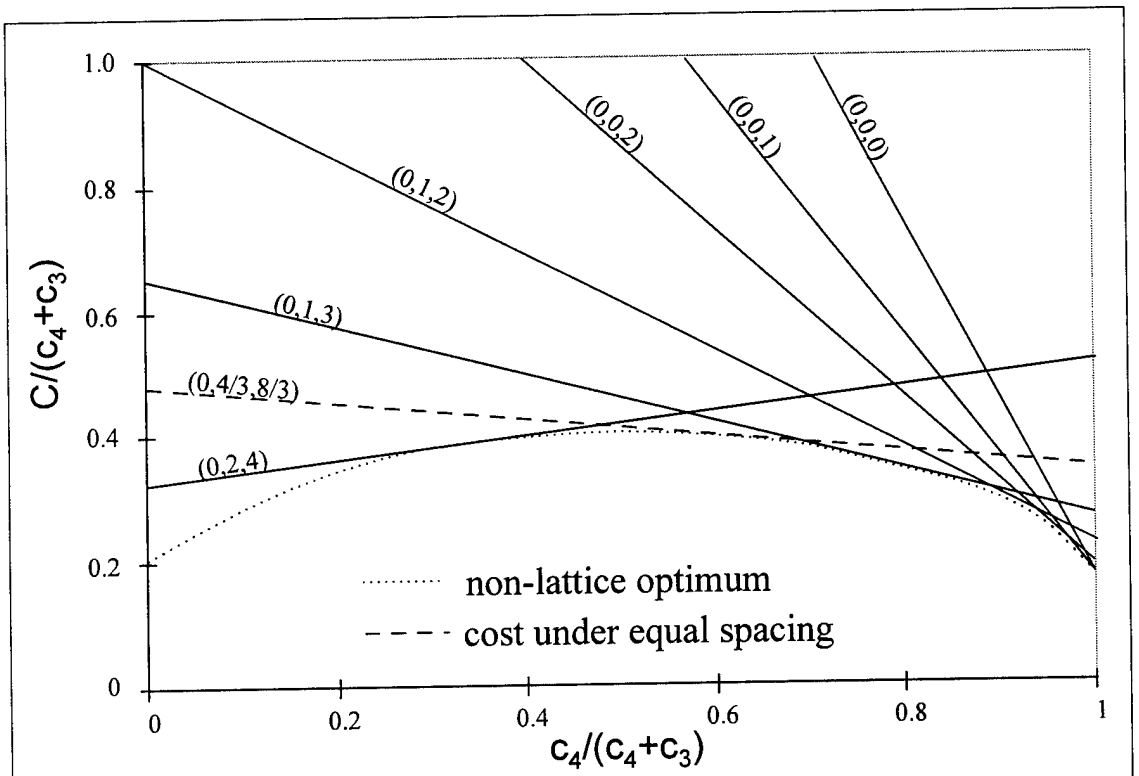


Figure 7. Plot of cost *vs.* coefficient ratio. Normalized cost *vs.* normalized overtime cost coefficient for several schedules with 5 slots and 3 customers. The dotted line is the optimal cost if the schedule is not constrained to be lattice. The dashed line is the cost if the interarrival times are all set equal.

The actual values of the waiting time and overtime coefficients are also irrelevant to the choice of optimal schedule; only their ratio affects the optimum. Figure 7 displays the total schedule cost for each possible c_3 ($= c_2$) and c_4 by normalizing c_4

and C by the factor $c_4 + c_3$. Only the six schedules (out of 15 possible) that become optimal for some choice of $c_4/(c_4 + c_3)$ (*i.e.*, are non-dominated) are displayed.

The schedule $\tau = (0, 2, 4)$ is seen to be optimal for values of $c_4/(c_4 + c_3) \leq 0.57$, with $\tau = (0, 1, 3)$ optimal for values of $c_4/(c_4 + c_3)$ between 0.57 and 0.86, and (0,1,2), (0,0,2), (0,0,1), and (0,0,0) becoming optimal in turn as the overtime cost increases further still. For any problem, the set of non-dominated schedules may be determined similarly over the range of $c_{N+1}/(c_{N+1} + c_N)$.

As the size of the lattice is decreased, the polygonal envelope of optimal solutions over the range of cost coefficients will converge pointwise to the optimal solution in the continuous case. This can be seen by creating a series of functions $f_i(c_{N+1}/(c_{N+1} + c_N))$, in which the i^{th} function is the optimal solution for the division of the schedule into $i + 1$ lattice points. The sequence $f_i(x), f_{2i}(x), f_{4i}(x)...$ is monotonic, nonincreasing, and bounded below by zero for each value of x. The sequence of functions therefore converges pointwise, and its limit must be the optimal solution in the continuous case. Each function in the series is the pointwise minimum of a collection of functions that are linear, so each is concave with respect to $c_{N+1}/(c_{N+1} + c_N)$, from which it follows that convergence of the sequence is uniform [134: Theorem 10.8]. Continuity of the cost function for continuous arrival times follows immediately. It should be noted that this argument applies equally when the unit waiting costs are not all equal; the scaling factor $(c_{N+1} + c_N)$ could just as easily have been $(c_{N+1} + c_{N-1})$, for example.

The importance of continuity lies in sensitivity analysis; a small modification in the importance of one of the cost terms will not result in a disproportionate improvement or degradation in the optimal cost. Appendix D explores the sensitivity and dependence of the optimal schedule and cost on various schedule parameters.

Section 3.1 showed the cost to be a function of integrals that, by Leibniz's Theorem, are differentiable over τ if the service PDFs are all of bounded variation [4]. For the cost function in Equation (1), they are also differentiable with respect to c.

Let $\tilde{C}(c)$ represent the optimal cost over possible schedules for a given unit cost vector c. Because $\tilde{C}(c)$ is the limit of a uniformly convergent sequence of concave, differentiable functions, it is also differentiable with respect to c [134: Theorem 25.7 and discussion preceding].

The straight dashed line represents the solution when the interarrival times are all set equal. This represents the traditional approach to schedule creation, and in this case it is close to the optimum for coefficient ratios near 0.6. It was shown in Section 3.7 that this traditional schedule usually is not optimal for a finite number of identical customers, but it was not clear how much savings could be achieved. If Bailey's dictum that "a doctor's time is 37.5 times more valuable than the patients' " [7] is followed, $c_4/(c_4 + c_3) = 0.974$, leading to the optimal schedule (0,0,1) for the chosen lattice size. The cost of this coarse lattice optimum is very close to that of the continuous optimum, but 40% less than that of the ubiquitous equi-spaced schedule, regardless of the number of schedule slots. It is clear that schedule optimization can lead to substantial cost improvements, even when fixing the sequence of customers.

Rather than an equi-spaced schedule, Charnetski [21], Weiss [164], and others sought the schedule that balances expected waiting times of each customer (except the first). Such a schedule would have the advantage that no customer would perceive a benefit to choosing a particular position in the customer sequence (other than the first appointment of the day). As it stands, many customers perceive a waiting-time advantage to being scheduled early in the customer sequence [13], and this perception is frequently correct. To give an example, for a 3-customer, 101-slot problem with $\Delta = 0.3$, the overtime point equal to the horizon, all cost coefficients equal to 1.0, all show probabilities equal to 1.0, and all services iid Exp(1.0), the results are shown in Table 2.

The cost of the schedule that comes closest to balancing waiting times in this case exceeds that of the globally optimal schedule by 23%. The end user must determine

Table 2. Comparison of results when balancing waiting times to results when minimizing the sum of waiting times. This is a 3-customer, 101-slot example with $\Delta = 0.3$, $\tau_v = \tau_h = 30.0$, $c_2 = c_3 = c_4$, and all services exponential with mean of 1.0.

	τ_1	τ_2	τ_3	W_2	W_3	overtime	cost
optimal	0	11.1	24.6	0.330	0.460	1.003	1.79
balanced	0	3.0	16.2	0.719	0.728	0.735	2.20

whether a near-balanced schedule would reduce customer or server dissatisfaction and whether this is a reasonable price to pay to do so.

This effort will concentrate on the globally optimal solution. A number of propositions require proof before proceeding with the proposed algorithm, beginning with the issue of convexity.

4.1 Convexity of the Cost Function

Wang considered a linear cost function of the expected waiting times and total service time for which $\gamma_1 = \cdots = \gamma_N = 1$, the scheduling horizon is unconstrained, and all cost functions are linear [160]. He proved the cost function under these conditions satisfies strong stochastic convexity with respect to the interarrival vector, $[\ \tau_2,\ \ \tau_3 - \tau_2,\ \ \tau_4 - \tau_3,\ \ \cdots,\ \ \tau_N - \tau_{N-1}\ ]$, and service vector. Shantikumar *et al.* define a function to be strongly stochastically convex if (in essence) it is convex almost surely. They pointed out that strong stochastic convexity implies stochastic convexity [143], and Wang used this result to obtain stochastic convexity for his cost function. In this section, Wang's argument is modified slightly to prove convexity of $C(\tau)$ for the cost function proposed in Equation (1) with respect to τ.

Theorem 1 *Assume the elements of τ are independent of each other, with the exception that $0 \leq \tau_1 \leq \tau_2 \cdots \leq \tau_{N+1}$, and τ_{N+1} may or may not be fixed. Assume the elements of χ are independent of each other and of τ. Then $C(\tau) = \sum_{i=2}^{N+1} c_i E[W_i(\tau)]$ is a convex function of τ and χ for $j \leq N+1$.*

Proof: Arbitrarily fix the service vector χ and the no-show vector θ. A recursive argument will be used to show that if W_j, conditioned on whether customer j showed, is convex, then so is W_{J+1}, conditioned on whether customers j and $j+1$ showed. The starting point will be $j=1$, since $W_1 = 0$ is convex with respect to τ, regardless of the value of θ_1. ∎

Equation (2) will now be modified to account for no-shows. As was done in Section 3.1, just for the purposes of calculating $W_{j+1}|_{(\theta_{j+1}=1)}$, one can picture that customer j always showed, waited for service, and then was served instantaneously with probability $1-\gamma_j$, otherwise undergoing service of length χ_j.

$$\begin{aligned} W_{j+1}|_{(\theta_{j+1}=1,\theta_j=1)} &= \max\left[0, W_j|_{(\theta_j=1)} + \chi_j - \tau_{j+1} + \tau_j\right] \\ W_{j+1}|_{(\theta_{j+1}=1,\theta_j=0)} &= \max\left[0, W_j|_{(\theta_j=1)} - \tau_{j+1} + \tau_j\right] \\ \Rightarrow W_{j+1}|_{(\theta_{j+1}=1)} &= \gamma_j \max\left[0, W_j|_{(\theta_j=1)} + \chi_j - \tau_{j+1} + \tau_j\right] \\ &\quad + (1-\gamma_j)\max\left[0, W_j|_{(\theta_j=1)} - \tau_{j+1} + \tau_j\right] \end{aligned} \tag{18}$$

Since $\max(x)$ is a convex, nondecreasing function, Equation (18) is a convex function of τ and χ if $W_j|_{(\theta_j=1)}$ is a convex function of τ and χ [134: Theorem 5.1]. It follows by mathematical induction that $W_j|_{(\theta_j=1)}$ is convex for all j and for fixed θ and χ.

Since $W_j = \gamma_j\left(W_j|_{(\theta_j=1)}\right)$, it also is convex. $E[W_j(\tau)]$ may be thought of as a convex combination of the W_j for each possible combination of χ and θ, if $\int_{\Re} W_j(\chi) df_j(\chi_j)$, the Riemann-Stieltjes integral defining the expectation of the j^{th} waiting time, exists for each j. This integral exists and is continuous with respect to τ if each of the service PDFs, $f_j(\chi_j)$, are of bounded variation and if $W_j(\tau)$ is continuous with respect to χ and τ [4: Theorem 7.38]. Continuity of $W_j(\tau)$ is assured over $\chi \in \Re^N$ and over $0 = \tau_1 \leq \tau_2 \cdots \leq \tau_N \leq \tau_{N+1}$, since Equation (18) holds and is dependent on the maxima of a set of continuous functions of τ and χ. Thus, the restriction is that each $f_j(\chi_j)$ be of bounded variation. If they are, then $E[W_j(\tau)]$ is

a convex function of τ, for all τ that maintain the order of scheduled arrivals. Note that $E[W_j(\tau)]$ is not stochastic, so stochastic convexity need not be invoked.

If the cost is a convex, nondecreasing function of each $E[W_j(\tau)]$, then $C(\tau)$ is also convex with respect to τ [134: Theorem 5.1]. In Equation (1), the cost is taken as a (multiple of a) convex combination of each $E[W_j(\tau)]$, so the theorem is proved. ∎

4.2 Modification of Simeoni's Approach

Simeoni examined the case of lattice arrival times, iid Erlang services, equal weightings for each customer's expected waiting time, a fixed time horizon, and without the possibility of no-shows. He proposed an efficient coordinate search algorithm that is extended here to include independent, general service distributions, cost functions that are convex combinations of the expected waiting times, and allowance of no-shows.

Lemma 2 *Consider an arbitrary schedule S_1. Create S_2, in which all customers arrive at the same times as in S_1, with the exception that customer i arrives at $\tau_i + \delta_i$ instead of τ_i. Select $j > i$, and create S_1' (S_2'), in which all customers arrive at the same times as in S_1 (S_2), with the exception that customer j arrives at $\tau_j + \delta_j$ instead of τ_j. Assume $\delta_i \leq \tau_{i+1} - \tau_i$ and $\delta_j \leq \tau_{j+1} - \tau_j$, so that service order is the same in each of the four schedules (shown schematically in Figure 8). Then*

$$C(S_1) + C(S_2') \leq C(S_2) + C(S_1') \tag{19}$$

Proof: Arbitrarily fix the service vector χ and the no-show vector θ. Since S_1 and S_1' (S_2 and S_2') differ only by the arrival time of customer j, it follows that

$$W_n(S_1) - W_n(S_1') = W_n(S_2) - W_n(S_2') = 0, \quad n = 2, 3, \ldots, j-1 \tag{20}$$

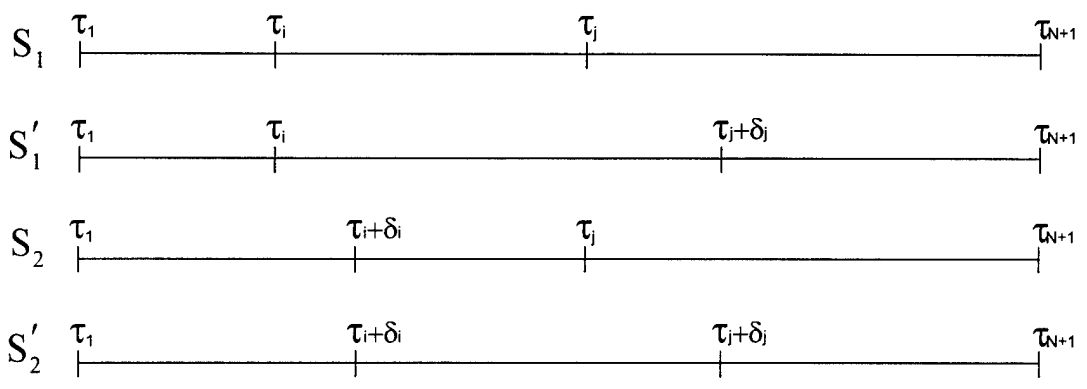


Figure 8. Arrival time schema for Lemma 2

It is apparent that $W_j(S_2) \geq W_j(S_1)$, since idle server time on the interval $[\tau_i, \tau_i + \delta_i]$ under schedule S_2 could be productively employed by customer i under S_1. Likewise, $W_j(S_2') \geq W_j(S_1')$. It follows that there is potentially more idle server time on the interval $[\tau_j, \tau_j + \delta_j]$ under S_1 than on the same interval under S_2. Hence, the decrease in waiting time for customer j realized by changing from S_1 to S_1' can be no greater than the decrease realized by changing from S_2 to S_2'. Furthermore, the increase in waiting time for any customer $n > j$ realized by changing from S_1 to S_1' must be at least as great as the increase realized by changing from S_2 to S_2'. Therefore,

$$W_n(S_1) - W_n(S_1') \leq W_n(S_2) - W_n(S_2'), \quad n = j, \ldots, N+1 \tag{21}$$

Since the above equations are true for each possible combination of χ and θ, and $E[W_n(S)]$ is just a linear combination of these possibilities,

$$E[W_n(S_1)] + E[W_n(S_2')] \leq E[W_n(S_2)] + E[W_n(S_1')], \quad n = j, \ldots, N+1, \tag{22}$$

and the desired result follows.[1] ∎

[1]This proof is presented in Vanden Bosch, Dietz, and Simeoni [158] and is due mainly to Dietz.

Lemma 2 is proved for continuous arrival times – *i.e.*, δ_i and δ_j may take on any positive values that maintain the same order of scheduled arrivals. In the remainder of this section, for clarity of exposition, it is assumed that arrival times are constrained to the evenly spaced lattice points $k\Delta$, with k positive integral and Δ positive real. Later, an algorithm will be presented that approximates the optimal schedule when arrival times are not restricted to lattice points.

The vector function $C(\tau)$ is called submodular if

$$C(x \vee y) + C(x \wedge y) \leq C(x) + C(y) \; \forall x, y \tag{23}$$

Define $x \vee y$ and $x \wedge y$ as the component-wise maximum and minimum of x and y, and define piecewise submodularity to be submodularity over the range of x and y that retain the same component orderings of the two vectors.

Theorem 3 *$C(\tau)$ is submodular over $\tau_1 \leq \tau_2 \cdots \leq \tau_{N-1} \leq \tau_N$.*

Proof: Let J be a set of the customers whose scheduled arrivals in S_1 are shifted later in S_1'. Let I be a set of customers whose scheduled arrivals in S_2 (S_2') are shifted later than in S_1 (S_1'). Without losing generality, let $I \bigcap J = \emptyset$. Since $S_1' \wedge S_2 = S_1$ and $S_1' \vee S_2 = S_2'$, the goal is to show that, for any choice of S_1, I, J, and the size of each individual shift,

$$C(S_1) + C(S_2') \leq C(S_1') + C(S_2) \tag{24}$$

If either I or J are empty, Equation 24 follows trivially, with equality holding. If I and J are each of cardinality 1, the theorem is simply a restatement of Theorem 2. For other cases, proceed recursively.

Assume the statement is true for sets I and J, each of cardinality less than or equal to N_I. Choose I and J of cardinality N_I and N_J, respectively, with $N_J \leq N_I$.

Then

$$C(S_1) - C(S_1') \leq C(S_2) - C(S_2') \tag{25}$$

Arbitrarily choose a customer h, and create S_3 (S_3') from S_2 (S_2') by shifting customer h's arrival later by some (equal) amount that does not alter customer arrival order. Let the list H consist of the single customer h. Again by the assumption,

$$C(S_2) - C(S_2') \leq C(S_3) - C(S_3') \tag{26}$$

Combination of the above equations yields

$$C(S_1) - C(S_1') \leq C(S_3) - C(S_3'). \tag{27}$$

This is the statement of Equation 24 for list I of cardinality $N_I + 1$ and list J of cardinality less than or equal to N_J. The same approach may be used to increase the cardinality of J and show the theorem holds if both I and J are of cardinality less than or equal to $N_I + 1$. Thus, the proposition is proved for all I and J by mathematical induction, and submodularity is established. ▪

Call the binary vector relation $\preceq$ "earlier than" and define it as follows: $x \preceq y$ if and only if vectors x and y are both the same length (call it $N+1$) and $x_i \leq y_i \; \forall i : i \in [0, N+1]$. If, in addition, $x \neq y$, then $x \prec y$. The relations $\succeq$ and $\succ$ are defined similarly.

The following two theorems are the heart of the scheduling algorithm to be proposed, allowing efficient fathoming of the solution space. Fathoming is defined as ensuring that some region of the solution space does not contain the optimal solution.

Theorem 4 *Suppose there is a schedule S_2 that fathoms all later schedules that maintain the same customer order. Suppose that $S_1 \preceq S_2$ and that $C(S_1) \leq C(S_2)$. Then S_1 fathoms all later schedules that maintain the same customer order.*

Proof: Shift the arrivals of an arbitrary set of customers in S_2 to arbitrary later times (without disturbing customer order) to create S_2'. Create S_1' from S_2' by shifting the arrivals of each customer by $S_1 - S_2$. Then by submodularity, $C(S_1) - C(S_1') \leq C(S_2) - C(S_2')$. However, $S_2 \preceq S_2'$, so $C(S_2) - C(S_2') \leq 0$, and it follows that $C(S_1) - C(S_1') \leq 0$. Therefore, S_1 fathoms all later schedules. ▪

The need for Theorem 3, over and above Lemma 2, is easy to overlook [145]. An example makes its importance clear. Suppose that

$$S_2 = [0 \quad 2 \quad 3 \quad 4]$$
$$S_1 = [0 \quad 1 \quad 3 \quad 4]$$
$$S' = [0 \quad 1 \quad 4 \quad 5]$$

Since it is not true that $S_2 \prec S'$, it is not clear whether S_2 fathoms S', given the assumptions of Theorem 4. Suppose that $C(S_1) < C(S_2)$. Theorem 2 ensures that S_1 fathoms later schedules that can be constructed from S_2 by moving one arrival time later and one earlier, but construction of S' requires moving one arrival time earlier and two arrival times later. Despite the fact that $S_1 \prec S'$, there is no assurance that $C(S_1) < C(S')$ without Theorem 3.

Theorem 5 *Suppose there is a schedule S_2 that fathoms all earlier schedules that maintain the same customer order. Suppose S_2' is formed from S_2 by shifting the arrival time of customer j an amount Δ later and that $C(S_2') \leq C(S_2)$. Then S_2' fathoms all earlier schedules that maintain the same customer order.*

Proof: The proof parallels that of Theorem 4. ▪

The last two theorems immediately suggest a strategy for finding the optimal schedule. Let K be the number of time slots in the schedule.

Fixed-Lattice Algorithm

1. Establish an early incumbent schedule, S_E. If no better bound is available, use $S_E = [\ 0 \quad 0 \quad \cdots \quad 0\ ]$.
2. Let m be the largest integer for which it holds that customer m in S_E is not scheduled at $t = (K-1)\Delta$.
3. Establish a candidate early schedule S by shifting the arrival time of customer m in S_E one time slot (Δ) later, unless this shift causes the order of customer arrivals to change. If all customers but the first are scheduled at $t = (K-1)\Delta$, stop. (Recall that the first customer's arrival time is fixed at 0.)
4. If $C(S) < C(S_E)$, let $S_E = S$ and return to step 2.
5. If $m > 2$, decrement m and return to step 3. Otherwise, each customer of the current S_E has shifted without improvement, and S_E is fixed.

The algorithm for establishing S_L, the late incumbent schedule, parallels the above. In finding S_L, if no better initial late bound is available, one should use the latest possible feasible schedule, in which $\tau_1 = 0$ and all other scheduled arrival times are at τ_{N+1}. However, once S_E is found, Theorem 8 below will provide a far more efficient initial bound for S_L.

The example in Table 3 applies the algorithm twice to a scheduling problem with three customers and six slots. In nine evaluations, S_E is found to be [0,1,3], and in eight more (two of which have already been evaluated), S_L is found to be [0,2,4], with a slightly lower cost than S_E.

A unique property of the algorithm is seen in the decision at iteration 6 of the early algorithm to abandon an apparently profitable search direction. A substantial cost reduction had just been achieved by shifting the second customer one slot, but the algorithm does not continue to shift the second customer, as most search

strategies would do. This apparent inefficiency is necessary in order to ensure all earlier schedules are fathomed.

The early and late fixed-lattice algorithms together fathom most of the solution space for this example. However, it is not yet proved that [0,2,4] is the global optimum, since [0,1,5] has not been evaluated and is neither earlier than S_E nor later than S_L. Three additional propositions will prove that $S_E \preceq \hat{S} \preceq S_L$, where $\hat{S}$ is the optimal lattice schedule, and that for sufficiently small lattices, S_E and S_L are quite close.

Lemma 6 *If* $S_1 \prec S_E$ *implies* $C(S_E) < C(S_1)$, *and* $S_2 \succ S_L$ *implies* $C(S_L) < C(S_2)$, *then* $S_E \preceq S_L$.

Proof: It is always possible to form S_E from S_L by shifting a list of scheduled arrival times I by Δ earlier and a list J by Δ later, $I \bigcap J = \emptyset$. It is also always possible both to form S_1 from S_E by shifting the scheduled arrival times of the customers in J by Δ earlier and to form S_2 from S_L by shifting the scheduled arrival times of the customers in I by Δ later. Then by submodularity, $C(S_1) - C(S_E) \leq C(S_L) - C(S_2)$. However, since S_E and S_L fathom earlier and later schedules, $C(S_1) - C(S_E) > 0$ and $C(S_L) - C(S_2) < 0$. The contradiction can be resolved only if either I or J is empty. This implies that either $S_E \preceq S_L$ or $S_L \preceq S_E$. It is impossible to have $S_L \prec S_E$, since the two schedules would then fathom each other, so $S_E \preceq S_L$. ∎

Theorem 7 $S_E \preceq \hat{S} \preceq S_L$.

Proof: Let S' be the lowest-cost schedule in the lattice of size Δ such that $S_E \preceq S' \leq S_L$. S' fathoms all schedules that are earlier than S_E or later than S_L, as well as those between S_E and S_L. Then if $S' \neq \hat{S}$, it must be that $\hat{S}$ contains some list I of customers whose scheduled arrival times are earlier than their scheduled arrival times in S_E. Likewise, $\hat{S}$ must contain some list J of customers whose scheduled arrival times are later than their scheduled arrival times in S_L. Form S_1 from S_E

Table 3. Determination of optimal early and late schedules

Early schedule algorithm

iteration	schedule	cost	improvement?
1	[0, 0, 0]	3.762	Y
2	[0, 0, 1]	2.819	Y
3	[0, 0, 2]	2.163	Y
4	[0, 0, 3]	1.854	Y
5	[0, 0, 4]	1.903	N
6	[0, 1, 3]	1.195	Y
7	[0, 1, 4]	1.215	N
8	[0, 2, 3]	1.216	N

Late schedule algorithm, starting at latest feasible schedule

iteration	schedule	cost	improvement?
1	[0, 5, 5]	3.525	Y
2	[0, 4, 5]	2.137	Y
3	[0, 3, 5]	1.623	Y
4	[0, 2, 5]	1.537	Y
5	[0, 1, 5]	1.766	N
6	[0, 2, 4]	1.069	Y
7	[0, 1, 4]	1.215	N
8	[0, 2, 3]	1.216	N

by shifting the arrival times of customers in I earlier by Δ, and form S_2 from S_L by shifting the arrival times of customers in J later by Δ. It follows that $S_1 \prec \hat{S} \prec S_2$ and $S_1 \prec S' \prec S_2$. Then, by Theorem 3,

$$C(S_1) + C(S_2) \leq C(S') + C(\hat{S}). \tag{28}$$

Since $S_1 \prec S_E$, $C(S_E) < C(S_1)$, and since $C(S') \leq C(S_E)$ by definition, then $C(S') < C(S_1)$. Also by definition, $C(\hat{S}) \leq C(S_2)$, so

$$C(S_1) + C(S_2) > C(S') + C(\hat{S}). \tag{29}$$

The contradiction between Equations (28) and (29) can only be resolved by concluding $\hat{S}$ cannot be distinct from S', which lies between S_E and S_L. ▪

Lemma 6 and Theorem 7 guarantee that if the inequality at step 4 of the fixed-lattice algorithm is strict, then $S_E \preceq \hat{S} \preceq S_L$. If, at some point, equality is observed at step 4 for each possible direction of improvement, the algorithm will stop. This is appropriate for convex functions such as the proposed cost function, since a "flat spot" implies that the optimum cost has been reached; if $C(S_E) = C(S_L)$, all schedules between S_E and S_L are optimal.

Remarkably, the algorithm as presented so far will also obtain bound the optima of a non-convex submodular function by S_E and S_L. Of course, the quality of the bounds depends on the nature and location of the nonconvexities; two local minima far apart ensure the bounds will also be far apart, since the algorithm cannot "pass through" such areas. Topkis proved that the set of minima of a submodular function defines a sublattice, and that result could be useful in the search for minima within these bounds [155, 156].

The following theorem establishes the efficacy of the bounds obtained only for convex functions, such as the proposed cost function.

Theorem 8 *Suppose* $C(S_E) \neq C(S_L)$. *Further suppose that either no two customers in* S_E *or that no two customers in* S_L *share the same arrival time. Then for each* j, τ_j *differs between* S_E *and* S_L *by at most* Δ.

Proof: Assume the opposite of the conclusion; suppose the scheduled arrival time of customer i had to be shifted more than Δ from its scheduled arrival time in S_E to reach its corresponding place in S_L. Let S_1 be the schedule formed from S_E by shifting i one time slot later. (The second condition of the theorem ensures this shift does not create an infeasible schedule.) It must be that $C(S_1) \geq C(S_E)$, or else the algorithm for obtaining S_E would have found S_1 as the early incumbent schedule. Likewise, $C(S_1) \geq C(S_L)$. Since $S_E \prec S_1 \prec S_L$, convexity is violated if $C(S_1) > C(S_E)$ or $C(S_1) > C(S_L)$. Therefore, $C(S_E) = C(S_1) = C(S_L)$. This proves the contrapositive of the theorem, so the theorem itself is also true. ▪

It may be that $C(S_E) = C(S_L)$ for a convex function. If τ_i differs in S_E and S_L by more than Δ for some i, it must be that the function is "flat" between these bounds with respect to τ_i, since otherwise the algorithm would have continued to find a better S_E or S_L. The situation $C(S_E) = C(S_L)$ is exceedingly rare and easily recognized and remedied when it occurs, so it is of minimal concern. In all other cases, the algorithm has been proved to produce tight bounds on the optimum for functions that are convex and for which Theorem 2 holds.

Normally, there is no problem when two customers occupy the same slot; the conclusion still holds. However, there is the remote possibility in some circumstances that in the search for S_E, a suboptimal schedule will be encountered in which two adjacent customers occupy the same slot and for which any shift results in a higher cost. This happens when the lattice size is so extremely coarse that a single shift of the latter customer traverses the entire region of improvement and ends up on the "opposite side" of the convex surface, regardless of the positions of subsequent customers. When the latter customer is shifted, the cost increases, and when the former is increased, an infeasible schedule ensues, forcing the search to fix the arrival

times of these two customers prematurely. Since the same may happen for S_L at a very different schedule, the conclusion of Theorem 8 no longer is true, and the number of schedules to be fathomed can be quite large.

When two customers are co-scheduled in S_E, a straightforward test to see if the above problem pertains is to shift all customers 2Δ later (if feasible) instead of Δ and search for S_L from this point. If the two customers in question are "glued", S_E and S_L will differ by more than Δ for these customers. Otherwise, the optimum can be obtained normally, at a cost of at most $N-1$ schedule evaluations. This procedure is incorporated into the program implementing the fixed-lattice algorithm provided in Section H.1, and the user is alerted to any problem.

If a problem is indeed ascertained, a reasonable approach is to meld these two customers into a single customer and restart the search for S_E at the point that the search stopped. This approach has not been automated and is left to the user's control.

It is emphasized that this co-scheduling problem has only been encountered when using lattice sizes on the order of or larger than the customer mean, which is unlikely to arise in actual situations.

As noted above, Theorem 8 provides an effective bound for S_L once S_E is found (or vice-versa). Instead of starting the search for S_L at the latest possible schedule, one can start at the schedule formed by shifting each of the arrival times of S_E one unit later, if feasible. (It is never feasible to leave the first schedule slot empty or to shift an arrival time past the schedule horizon.) If it is suspected that the optimum schedule is closer to the latest possible schedule than to the earliest, one may reduce the number of iterations required by finding S_L first, then finding S_E.

The fixed-lattice algorithm is a sort of cyclic coordinate search, in which a series of tests for improvement are performed along the coordinate axes in a specified order. However, as noted earlier, if there is an improvement, the new candidate optimum

is immediately accepted, and the search in that direction is halted in favor of a new direction. Such a search method seems counterintuitive; it seems reasonable to persist in a search direction for as long as an improvement is realized, as many nonlinear programs (NLPs) do. The reason the search is halted is that Theorems 4 and 5 are of help in fathoming solutions only if changes of Δ are made at each step. That this search algorithm is often an improvement over other approaches (*cf.* Appendix C) is partly due to the increase in schedule cost as customers are scheduled close together during the optimization process; a standard NLP approach such as Hooke-Jeeves cannot achieve much improvement over the cost function by continuing for long in any single direction, limiting the effectiveness of any line search it employs.

4.3 Fixed-Lattice Examples

A better understanding of the problem and solution algorithm can be gained by considering some examples. Let $I(S_E)$, $I(S_L)$, and $I(\hat{S})$ be the number of iterations (schedule cost evaluations) required by the early, late, and enumeration phases of the algorithm.

The example from the beginning of this chapter is considered first, in which there are 5 lattice points (slots), 3 customers, and services are iid exponential distributions with mean of 1. Overtime commences at the schedule horizon. The optimal schedules were already obtained by exhaustive enumeration of the 15 possibilities. For the specific case of $c_2 = c_3 = c_4$, the lattice algorithm produces

$$S_E = [\ 0\ \ 1\ \ 3\ ]\quad C(S_E) = 0.9170\quad I(S_E) = 8$$
$$S_L = [\ 0\ \ 2\ \ 4\ ]\quad C(S_L) = 0.8369\quad I(S_L) = 2$$
$$\hat{S} = [\ 0\ \ 2\ \ 4\ ]\quad C(\hat{S}) = 0.8369\quad I(\hat{S}) = 0$$

Here, S_E was found using the fixed-lattice algorithm, halting after 8 iterations. A candidate S_L was formed by shifting the second and third customer arrival times one slot. The algorithm evaluated two additional schedules, [0 1 4] and [0 2 3], but no improvement was obtained. Since S_L and S_E differed, the schedules between them had to be exhaustively evaluated. However, there are only two schedules between them, and these already were evaluated during the calculation of S_L. Thus, no new schedules were evaluated during the enumeration phase, and $\hat{S} = S_L$.

Figure 7 shows that [0 1 3] and [0 2 4] are equal in cost when $c_4/(c_4+c_3) \approx 0.569$. It turns out that S_E and S_L differ by two customers if $c_4/(c_4+c_3) \in [0.45, 0.62]$. For $c_4/(c_4+c_3) \in [0, 0.45] \bigcup [0.62, 0.92]$, $S_E = S_L$.

For this small example, the algorithm provides little improvement over full enumeration of the schedules. Extending the number of schedule slots to 10 and the number of customers to 11 (while maintaining the same service mean) produces

$$S_E = [\ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\ \ 9\ \ 9\ ] \quad C(S_E) = 16.26 \quad I(S_E) = 153$$
$$S_L = [\ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 6\ \ 7\ \ 8\ \ 9\ \ 9\ \ 9\ ] \quad C(S_L) = 16.02 \quad I(S_L) = 18$$
$$\hat{S} = [\ 0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 6\ \ 7\ \ 8\ \ 9\ \ 9\ \ 9\ ] \quad C(\hat{S}) = 16.02 \quad I(\hat{S}) = 6$$

Here, 177 schedules were evaluated out of the 92,378 possible, showing the potential of the fixed-lattice algorithm.

An extreme example with respect to computation is seen with 20 slots, 15 customers, $c_2 = \cdots = c_{16} = 1$, and an Erlang-4 service distribution with mean of 2Δ. The algorithm produces

$$S_E = [\ 0\ \ 1\ \ 3\ \ 5\ \ 7\ \ 9\ \ 11\ \ 13\ \ 15\ \ 17\ \ 19\ \ 19\ \ 19\ \ 19\ \ 19\ ]\Delta$$
$$C(S_E) = 51.06 \quad I(S_E) = 458$$

$$S_L = [\ 0\ \ 2\ \ 4\ \ 6\ \ 8\ \ 10\ \ 12\ \ 14\ \ 16\ \ 18\ \ 19\ \ 19\ \ 19\ \ 19\ \ 19\ ]\Delta$$
$$C(S_L) = 50.72 \qquad I(S_E) = 10$$
$$\hat{S} = [\ 0\ \ 2\ \ 4\ \ 6\ \ 8\ \ 10\ \ 12\ \ 14\ \ 16\ \ 18\ \ 19\ \ 19\ \ 19\ \ 19\ \ 19\ ]\Delta$$
$$C(\hat{S}) = 50.72 \qquad I(\hat{S}) = 492$$

In this case, the majority of the evaluations are consumed in resolving the small difference between S_E and S_L, eventually obtaining no improvement. It is very rare that S_E and S_L differ in nearly half their slots. Even with such a large enumeration phase, only 960 of the possible 818,809,200 schedules required evaluation.

Simeoni conjectured that $\hat{S}$, the lattice optimum, always coincided with either S_L or S_E [145]. A counterexample is the case in which 12 customers are to be scheduled into 10 slots. No-shows are not allowed, and services are iid exponential with mean equal to Δ. Costs are linear, overtime commences at the schedule horizon, and $c_2 = \cdots = c_{13} = 1$. For this case,

$$S_L = [0\ 1\ 2\ 3\ 4\ 5\ 6\ 8\ 9\ 9\ 9\ 9]\Delta \qquad C(S_L) = 21.31 \qquad I(S_L) = 108$$
$$S_E = [0\ 0\ 1\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 9\ 9]\Delta \qquad C(S_E) = 21.10 \qquad I(S_E) = 27$$
$$\hat{S} = [0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 9\ 9]\Delta \qquad C(\hat{S}) = 21.04 \qquad I(\hat{S}) = 20$$

$C(S_L)$ and $C(S_E)$ are quite close, preventing the algorithm from resolving the optimum until the 20 schedules between S_L and S_E are enumerated. Each of these enumerated schedules are also quite close in cost, but only $\hat{S}$ above is lower in cost than both S_E and S_L.

If S_E and S_L differ by k arrivals, there are 2^k schedules between them, inclusive. However, some of those schedules cause the order of arrivals to change, and others have already been evaluated in the process of finding S_E and S_L. The actual

number of schedules required to be enumerated is far less and is discussed in detail in Appendix C.

If S_E and S_L do differ, the likelihood is very high that the optimum is one of these two schedules (*cf.* Appendix C). For instance, in a series of 3000 optimizations of 10-customer, 21-slot schedules with parameters chosen randomly from a realistic set, each of the optima coincided with either S_E or S_L. When the optimum does not coincide with either S_L or S_E, it appears very likely that the optimum schedule differs by only two customers from either S_L or S_E, as in the counterexample. No case of the optimum differing from both S_E and S_L by more than two customers has been observed, although such a situation cannot be ruled out.

The above examples show that $C(S_E) \approx C(S_L) \approx C(\hat{S})$. Thus, in cases in which suboptimal solutions are acceptable, one may decide just to obtain S_L or S_E and use it as a suboptimal solution. One is assured by Theorem 8 that each arrival time in this approximation is within Δ of its optimum. In each of the rare cases observed in which the optimum was neither S_E nor S_L, the cost improvement obtained by selecting the global optimum over those schedules was less than 0.5%.

4.4 Algorithms for Finding the Optimal Fine-Lattice or Continuous Schedule

A natural extension to the fixed-lattice algorithm is first to apply the fixed-lattice algorithm using a coarse lattice in an effort to fathom schedules under a finer lattice more efficiently. Such an algorithm might also be employed to obtain increasingly better approximations to the unconstrained (continuous) optimum by setting the lattice size successively smaller. Such an algorithm was attempted by Simeoni [145]. His success was limited by the lack of a way to obtain efficient bounds that apply when the lattice size is altered. (That S_E and S_L are no longer necessarily bounds under a new lattice size is immediately apparent if one considers

the possibility that $S_E = S_L$.) This lack is remedied by the following four corollaries to Theorem 8.

Corollary 9 *Suppose S_E is obtained for arrival times constrained to a lattice of size Δ. Let $\hat{S}'$ be the optimum schedule for the same problem with lattice size Δ' such that Δ/Δ' is a positive integer. Let Ψ_{N+1} be a vector of $N+1$ elements, all equal to unity. Then $\hat{S}' \preceq S_E + \Delta\Psi_{N+1}$.*

Proof: For these particular values of Δ', $S_E + \Delta\Psi_{N+1}$ lies in the new lattice system, so $C(\hat{S}') \leq C\left(S_E + \Delta\Psi_{N+1}\right)$. In addition, $C(\hat{S}) \leq C\left(S_E + \Delta\Psi_{N+1}\right)$, where $\hat{S}$ is the optimum under lattice size Δ. If the conclusion is assumed false, then $\hat{S} \preceq S_E + \Delta\Psi_{N+1} \prec \hat{S}'$. However, in contradiction with Theorem 1, the cost function cannot be convex now, since a maximum exists on the line segment connecting $\hat{S}$, $S_E + \Delta\Psi_{N+1}$, and $\hat{S}'$, and that maximum is not at an endpoint. The assertion is thus proved indirectly. ▪

Corollary 10 *Suppose S_L is obtained for arrival times constrained to a lattice of size Δ. Let $\hat{S}'$ be the optimum schedule for the same problem with lattice size Δ' such that Δ/Δ' is a positive integer. Then $\hat{S}' \succeq S_L - \Delta\Psi_{N+1}$.*

Proof: Parallels that of Corollary 9. ▪

Corollaries 9 and 10 proved that if Δ' is an integral fraction of Δ, then $S_L - \Delta \preceq \hat{S}' \preceq S_E + \Delta$. The same bounds hold for the continuous optimum, $\tilde{S}$.

Corollary 11 $S_L - \Delta\Psi_{N+1} \preceq \tilde{S} \preceq S_E + \Delta\Psi_{N+1}]$.

Proof: Parallels those of Corollaries 9 and 10. ▪

If Δ' is not an integral fraction of Δ, the proofs for Corollaries 9 and 10 fail, since the relationship between the two optimum lattice schedules is unclear; in some cases, a smaller lattice leads to a larger optimal cost. A further relaxation of the search bounds is required.

Corollary 12 *Suppose S_E and S_L bound the optimum under lattice size Δ. Let $\hat{S}'$ be the optimum schedule for the same problem with lattice size Δ' such that $\Delta' < \Delta$. Then $S_L - (\Delta + \Delta')\Psi_{N+1} \preceq \hat{S}' \preceq S_E - (\Delta + \Delta')\Psi_{N+1}]$.*

Proof: By Corollary 11, S_E and S_L must lie at most Δ from $\tilde{S}$. Likewise, S'_E and S'_L, the bounds under lattice size Δ', must lie at most Δ' from $\tilde{S}$. Then S'_E lies at most $\Delta + \Delta'$ from S_L, and S'_L lies at most $\Delta + \Delta'$ from S_E. Since $S_E \preceq S_L$, and $S'_E \preceq S' \preceq S'_L$, the desired result follows. ▪

The rough bounds obtained when reducing the lattice size by an integral fraction are substantially better than those in the general case, and one should take advantage of this property whenever possible. However, sometimes the choices of lattice size are limited; one may have to resort to a non-integral reduction in lattice size. An algorithm for obtaining the exact optimum for a fine lattice size Δ follows.

Variable-Lattice Algorithm

1. Choose a coarse lattice size, Δ'. When feasible, make Δ' a multiple of Δ. Set S'_E, the lower bound on the optimum schedule under lattice size Δ', equal to $[0, 0, \cdots, 0]$.
2. Improve S'_E using the fixed-lattice algorithm, with the current value of S'_E as a starting point.
3. Choose Δ'' such that $\Delta \leq \Delta'' < \Delta'$. When feasible, choose Δ'' such that Δ' is a multiple of Δ'', or Δ'' is a multiple of Δ, or (preferably) both.
4. If Δ' is a multiple of Δ'', let $S''_L = S'_E + \Delta'\Psi_{N+1}$. Otherwise, let $S''_L = S'_E + (\Delta' + \Delta'')\Psi_{N+1}$. S''_L is an upper bound on the optimum schedule under lattice size Δ''.
5. Improve S''_L using the fixed-lattice algorithm, with the current value of S''_L as a starting point.

6. Choose Δ' such that $\Delta \leq \Delta' < \Delta''$. When feasible, choose Δ' such that Δ'' is a multiple of Δ', or Δ' is a multiple of Δ, or (preferably) both, to take advantage of the improvement in bounds under Corollaries 9 and 10.

7. If Δ'' is a multiple of Δ', let $S'_E = S''_L - \Delta''\Psi_{N+1}$. Otherwise, let $S'_E = S''_L - (\Delta'' + \Delta')\Psi_{N+1}$.

8. If $\Delta'' \neq \Delta$, go to step 2. Otherwise, stop; S'_E and S''_L are bounds to the optimal solution under Δ. Since $S''_L \preceq S'_E + \Delta\Psi_{N+1}$, the bounds are close, and an exhaustive search is feasible.

For an approximation to the unconstrained optimum, set Δ to the desired accuracy and apply the variable-lattice algorithm. Initial experiments indicate that, for obtaining the unconstrained optimum, the lattice size should be decreased at each step by a factor of at least four; smaller reductions do not appear effective at reducing the number of cost evaluations required. One should start with a lattice with at least four points within the time horizon, or else the added problem setup time will more than offset any savings in time. When using the algorithm to obtain the exact optimum for a fine lattice, these rules are mitigated by the better bounds obtained if integral reductions are chosen.

4.5 Variable-lattice Example

As an example, consider a problem with 5 customers and 301 schedule slots. The services are all Erlang(2) with mean of 1, and customers always show. The schedule horizon and the overtime point are 2 and the cost coefficients all equal to 1. The optimum schedule is [0 94 226 300 300], which takes 367 evaluations and 776 seconds to find using the fixed-lattice algorithm. One variable-lattice approach is shown in Table 4. Here, the approach is to start the fixed-lattice algorithm at $K - 1 = 8$ and quadruple $K - 1$ at each iteration of the variable-lattice approach for each iteration except the last. For the example, this approach took a total of

45.3 seconds, a reduction of 94% over the fixed-lattice approach. The total number of schedule evaluations required is 70, but time is a better measure of effectiveness here, since the time to perform a single evaluation is dependent on K.

Alternate paths to reach $K-1=300$ were tried, with (10, 60, 300), (15, 75, 150, 300), and (10, 50, 150, 300) taking slightly more time than (8, 32, 128, 300). These paths were tailored to the specific problem so that the reduction in Δ is an integer at each iteration. However, no improvement in run time was attained over the selected generic approach, suggesting that the improvement in bounds when an integer reduction is achieved is not worth pursuing. For comparable problem sizes, a fourfold reduction in Δ for each iteration is a reasonable value.

For larger problems, initial experiments suggest a reduction in Δ of a factor of 2 for each iteration is effective. For example, given the solution of the above problem for $K-1=300$, a subsequent single iteration to reach $K-1=1200$ takes 141 seconds, while taking the path (300,600,1200) takes $27+76=103$ seconds. An approach that has proved effective in practice is to switch from a fourfold to a twofold reduction in Δ when $K \geq 100$.

Table 4. Comparison of fixed- and variable-lattice results for a sample problem with $N=5$ and $K-1=300$. The schedules are in units of the current Δ.

K-1	starting schedule	optimal schedule	cost	evals	time
8	[0 0 0 0 0]	S_E: [0 2 6 8 8]	7.82195	7	2.63
32	[0 12 28 32 32]	S_L: [0 10 24 32 32]	7.80134	15	3.57
128	[0 36 92 124 124]	S_E: [0 40 97 128 128]	7.80133	24	9.18
300	[0 97 230 300 300]	S_L: [0 94 226 300 300]	7.80127	16	20.38
300	[0 93 225 299 299]	S_E: [0 94 226 300 300]	7.80127	8	9.55

Little improvement is attained in the optimum as the number of slots is increased. This is generally true, and is justification for limiting the lattice size in practical problems. This limits the usefulness of the variable-lattice algorithm. The reader should not get the impression from this that the cost function is flat; the cost

of the intuitively attractive equi-spaced schedule (*i.e.*, [0 60 120 180 240]Δ for $K-1=300$), is 8.87, 13% higher than the coarsest lattice solution tabulated.

4.6 The Dynamic Problem

The goal of this chapter has been to solve the static problem: What is the optimal schedule for a given sequence if it must be determined before the start of the schedule? In the dynamic problem, the schedule must be determined at some time t_d after the start, given that the realizations of the services to that point are known. This section solves this dynamic problem by transforming it to a static problem.

Two researchers have solved dynamic problems, both assuming exponential services. Wang solved a different dynamic problem than that above: if a new customer is added to the system at t_d, what should be its scheduled arrival time, given the scheduled arrival times of the other customers are fixed [160]? This is a realistic problem, in that it may not be possible to reschedule jobs, but an analytical treatment leads to little improvement in the cost when compared to the current practice of "just sticking it in somewhere". Liao solved the dynamic problem treated here for iid Erlang service times [97], but it is not possible to extend his solution even to idd Erlang services unless the current phase is known of each customer at t. This is seldom the case.

Suppose that at t_d, N_c customers have completed service and N_s are in service. The task is to schedule the remaining $N-N_c-N_s$ customers into the interval $[t_d, \tau_h]$. The customers who completed service obviously do not affect the optimal dynamic schedule. However, the ones in service pose the problem noted by Liao: if the number of phases completed thus far is not known, the memoryless properties of the exponential phases cannot be exploited. To employ the phase-type distribution

approach to cost calculation recommended in Chapter III, it is necessary to revise the state probability vector $p(t_d)$.

This is a relatively straightforward task, since t_s, the starting time of customer $N_c + 1$ (the customer in service at t_d), is known. Start the cost algorithm at t_s, with all the probability mass in the first phase of customer $N_c + 1$, and determine the probability vector at t_d. This probability vector is obviously incorrect, since it gives a nonzero probability of customer $N_c + 1$ and those subsequent having completed service at t_d, which is known at t_d not to be the case. However, this probability vector is easily transformed into the correct one, which is conditioned on customer $N_c + 1$ not having completed service at t_d. All that is necessary is to zero out the probabilities of states corresponding to customer N_c+1 having completed service and then renormalize so that the probabilities still sum to 1.0. Now the cost algorithm can be restarted and will provide correct expected waiting times for customers subsequent to customer $N_c + 1$. The optimization algorithms provided in this chapter obviously still work, since the dynamic problem is transformed by this artifice into a static problem. The fact that this new static process starts with a customer in service poses no difficulties.

Consider an example. Five customers each have Erlang(4) services with mean of 1.0, show probabilities of 1.0, and were to be scheduled into $[0, 5]$. Cost coefficients and overtime coefficient are 1.0. The optimal static schedule using a lattice size of 0.05 was [0.00 1.05 2.30 3.55 4.65], and the optimal cost was 1.885, of which 1.51 came from the overtime and waiting times of customers 3, 4, and 5. Suppose the first customer completed service before the second arrived and that at $t_d = 2.0$, the second is still in service.

The first task is to determine $p(2.0)$, given the knowledge above. At the start of customer 2's service, t_s, all the probability mass is concentrated in the first phase

of customer 2:

$$p(t_s) = [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 1.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad \cdots\ ]$$

Since customer 1 was not in service upon customer 2's arrival, $t_s = 1.05$, the arrival time of customer 2 in the optimal static schedule. Application of Equation(7) using the transition matrix from the original static problem and $\Delta = 0.95$ yields

$$p(2.0) = \begin{bmatrix} 0.000 & 0.000 & 0.000 & 0.000 & 0.022 & 0.085 & 0.162 & \cdots \\ \cdots & 0.205 & 0.194 & 0.148 & 0.148 & 0.094 & 0.051 & \cdots \\ \cdots & 0.024 & 0.010 & 0.004 & 0.001 & 0.000 & 0.000 & 0.000 \quad 0.000 \end{bmatrix}$$

(This row vector is displayed in three rows due to space limitations.) Since it is known that at 2.0, customer 2 is still in service, all states other than 5 through 8 must be zeroed and $p(2.0)$ must be renormalized, yielding

$$p(2.0) = [\ 0.0 \quad 0.0 \quad 0.0 \quad 0.0 \quad 0.047 \quad 0.179 \quad 0.341 \quad 0.432 \quad 0.0 \quad \cdots\ ]$$

The cost algorithm may now be started at 2.0 using this initial probability vector and the original transition matrix. Application of the fixed-lattice algorithm to this transformed problem yields the optimal dynamic schedule

$$[\ 0.00 \quad 1.05 \quad 2.45 \quad 3.65 \quad 4.75\ ]$$

which is slightly later than the original static schedule.

The cost contributed by the overtime and waiting times of customers 3, 4, and 5 in this dynamic schedule is 1.72. substantially more than the 1.51 given in the static schedule. This is to be expected, since the realization of customer 2 turned out to be larger than was expected at the start of the schedule. More appropriate is to compare this cost to that obtained if the static schedule had been kept, given

customer 2 had not arrived by t=2.0. This cost is 1.74, so improvement under the dynamic schedule was slight in this case. In situations where the realizations of services are far from their means, when the show probability is appreciably less than 1.0, or when customers are removed or added to the system, the dynamic solution is expected to be a substantial improvement over retaining the static solution.

4.7 Variations on the Scheduling Problem

The problem as defined may not seem relevant to some applications, due to certain features not modeled. However, the algorithm is robust enough to tolerate certain modifications. As one example, suppose a doctor required 30 minutes blocked out of his/her morning of seeing outpatients in order to perform surgery rounds. This could be modeled in several ways. If the duration and start of this period were flexible, one way to model it would be as an added patient with a mean service time of 30 minutes. If the duration and start of this period were fixed, and it was deemed unacceptable to have patients wait until the doctor returned, then it would be best to break the morning into two separate schedules.

Suppose the round times were fixed and it was deemed appropriate for patients to wait during rounds, even if the doctor had to leave in the middle of the consult. Then one could take advantage of the fact that the lattice algorithms work even when the amount of time between schedule slots is not constant. One could create a gap between two slots precisely equal to the duration of rounds. One would expect in this case to see a very different optimal patient sequence and schedule than in the previous treatments.

The service protocol so far has been first-come, first-serve (FCFS). If a priority or preemptive protocol obtain, the cost function is still submodular, and the lattice algorithms still function. Only the cost calculation would become more involved. Likewise, if the single server were replaced by a network of servers, the argument for the submodularity of the cost function is unchanged.

It is thus seen that the proposed algorithms are rather more robust than presented and are flexible enough to model a variety of features encountered in appointment systems.

V. *Determining the Optimal Sequence of Arrivals*

The previous chapter addressed the problem of determining an optimal schedule for a given sequence of arrivals. The task left is to determine which sequence will be optimal for a given problem. For small problems, this can be accomplished by exhaustive enumeration of all sequences, calculation of their optimal schedules, and selection of the best alternative. However, this approach quickly becomes unwieldy as the number of customer classes increases.

This chapter examines some characteristics of optimal sequences. The optimal sequence is seldom one in which customers are ordered by weighted means, weighted variances, or by any other simple measure. In fact, the optimal sequence frequently places identical customers at very different places in the schedule. It will be seen that this surprising behavior is inherent even in simple deterministic problems. There is evidence that even deterministic appointment sequencing problems are NP-hard. Several solution approaches to the stochastic problem are considered. These are abandoned in favor of a heuristic approach, which will be shown to be quite effective in obtaining optimal or near-optimal solutions in polynomial time.

5.1 Deterministic Examples

Deterministic problems are discussed in detail in Appendix A. Services are assumed to be deterministic and known, no-shows are not allowed, and the unit cost of overtime is zero. The results obtained in that appendix are used here to determine the optimal sequences for two sample deterministic problems. These problems will demonstrate that the curious features of stochastic sequencing problems have their root not in their stochastic nature, nor in the inclusion of overtime, but in the nature of the family of deterministic waiting problems at the heart of each stochastic problem. Consider the deterministic example defined by the parameters in Table 5. Customers are labeled A, B, and C to avoid confusion of indices with order.

Table 5. Parameters for the first deterministic example

customer	χ	c
A	3	8
B	2	3
C	1	1

When $\tau_h = 0$, the optimal schedule is trivially [0 0 0]. Theorem 13 in Appendix A proves that WSPT[1] yields the optimal sequence, ABC. The optimal cost is

$$C(\tau) = c_{\text{B}}(\chi_{\text{A}}) + c_{\text{C}}(\chi_{\text{A}} + \chi_{\text{B}}) = 15$$

Now set $\tau_h = 1$. As discussed in Appendix A, the optimal schedule becomes [0 1 1], regardless of sequence. Retaining the WSPT order yields a cost of

$$C(\tau) = c_{\text{B}}(\chi_{\text{A}} - 1) + c_{\text{C}}(\chi_{\text{A}} - 1 + \chi_{\text{B}}) = 10$$

But the optimal sequence can be shown by the methods of Appendix A to be CAB, for a cost of

$$C(\tau) = c_{\text{A}}(\chi_{\text{C}} - 1) + c_{\text{B}}(\chi_{\text{C}} - 1 + \chi_{\text{A}}) = 9$$

The optimal sequences and schedules for various values of τ_h are shown in Table 6.

Table 6. Optimal schedules and sequences for a deterministic example. This is the problem described in Table 5

horizon	sequence	schedule			cost
0	ABC	0	0	0	14
1	CAB	0	1	1	9
2	BAC	0	2	2	3
3	CBA	0	1	3	0
	BCA	0	2	3	0

[1] Weighted shortest processing time (WSPT) is the ordering of customers from smallest to largest value of χ_j/c_j. A process is called WSPT if the optimal sequence is always WSPT.

This dependence of the optimal sequence on τ_h is remarkable, but even more remarkable is the propensity for the optimal sequence to place identical customers in very different places. Suppose $N = 4$, $\chi = [3, 1, 1, 1]$, and $c = [4, 1, 1, 1]$. Call the first customer A and the other three (identical) customers B. The optimal results are shown in Table 7. This tendency for the optimal sequence to separate identical customers in some situations appears to be unique among single-machine deterministic sequencing problems examined in the literature.

Table 7. Optimal schedules and sequences for another deterministic example. Here, $\chi = [3, 1, 1, 1]$ and $c = [4, 1, 1, 1]$

horizon	sequence	schedule				cost
0	ABBB	0	0	0	0	12
1	BABB	0	1	1	1	7
2	BBAB	0	1	2	2	3
3	BBBA	0	1	2	3	0

5.2 Stochastic Examples

A service time distribution can be considered to be a convex combination of a number of deterministic services. It should therefore be no surprise that optimal solutions to stochastic problems display the same odd behavior that was shown for deterministic problems. While many stochastic scheduling problems have simple optimal ordering rules such as ordering by weighted shortest expected processing time (WSEPT) or by weighted shortest variance of the processing time (WSVPT), the following examples will help disabuse the reader of any remaining notion that some simple ordering principle exists for this problem. Optimal sequences in this section are obtained by optimizing the schedule for every possible sequence and choosing the best. In this example, suppose that there are two classes of customers, each with show probability of 1.0, and that each waiting cost and overtime cost is equally weighted. Customers have the service distributions shown in Table 8.

Table 8. Parameters for a stochastic example

class	service distribution	mean	variance	skewness
A	Erlang-2 with u=2.0	1.00	0.50	1.41
B	Cox-4 with b1=b2=b3=1.0, u1=1.3, and u2=u3=u4=13.0	1.00	0.61	1.91

Suppose there are two customers of class A and two of class B to be scheduled within τ_h. The sequences in Table 9 are found (by enumeration) to be optimal over a 101-slot lattice:

Table 9. Optimal solutions for the stochastic example

$\tau_h < 0.46$	costs of all sequences are equal
$0.46 < \tau_h < 1.14$	BBAA
$1.14 < \tau_h < 1.21$	ABAB or ABBA
$1.21 < \tau_h$	AABB

This example shows that, even when the cost coefficients, means, and show rates of all customers are equal, customers cannot merely be ordered by increasing service variance, as has been hypothesized in previous research [162, 164].

The reader may suspect the effect of higher moments is causing the reversal of customer order in those regions. However, the coefficients of variation for the two classes are 0.71 and 0.78, respectively, so the third moment should have little effect on the schedule, and thus on the sequence, of arrivals.

Consider a 3-customer problem with Erlang services and the parameters listed in Table 10.

Figure 9 maps the optimal sequence of this problem as the schedule horizon and the overtime unit cost are varied. The sequence CBA orders customers by WSEPT, while ABC orders customers by WSVPT. These are the most commonly encountered optimal schedules over the space depicted. (The notation Cxx indicates sequences CAB and CBA are very nearly identical in cost, and that one of them is optimal.) For a given unit cost vector, the optimal sequence typically changes from

Table 10. Parameters used in Figure 9

customer	unit cost	Erlang phases	phase rate	mean	variance
A	1.0	1	1.0	1.00	1.00
B	1.0	2	2.0	1.00	0.50
C	1.0	2	3.0	0.67	0.22

Cxx to CBA to ABC as the horizon increases. This is intuitively appealing, since one would expect the service mean to have more cost impact than the variance when the schedule is very constrained, and the probability that customers 2 through N must each wait is high. Conversely, when the schedule is much less constrained, and customers seldom have to wait, one would expect the service mean to play a very small part in sequence preference. That the transition point between optimality of the WSEPT and WSVPT sequences is at a higher τ_h when c_4 is small is also intuitively appealing, since variance of an earlier service is expected to have more impact than mean on the overtime.

This interface between optimality of the mean- and variance-ordered sequences is often marked by intermediate optima, as can be seen in the enlargement at the bottom of Figure 9. Of necessity, points near this border depict problems in which two sequence costs are very close, and thus small numerical instabilities in the cost evaluation process should be more evident. However, these numerical artifacts contribute only a small part to the jumble of optima at the border. The cost evaluation appears to be accurate to at least 0.001%, and most of the cost differences between sequences in this area are greater than 0.01%. While the shape of the border is only roughly accurate, there are indeed areas where sequences CAB and BAC are optimal for this problem.

Wang attempted to prove that if services are exponential and all unit waiting time costs are equal, then customers will be optimally ordered by decreasing service rate. This ordering is identical to both WSEPT and WSVPT in this situation. No counterexamples have been observed when the overtime point is at zero, as it is in

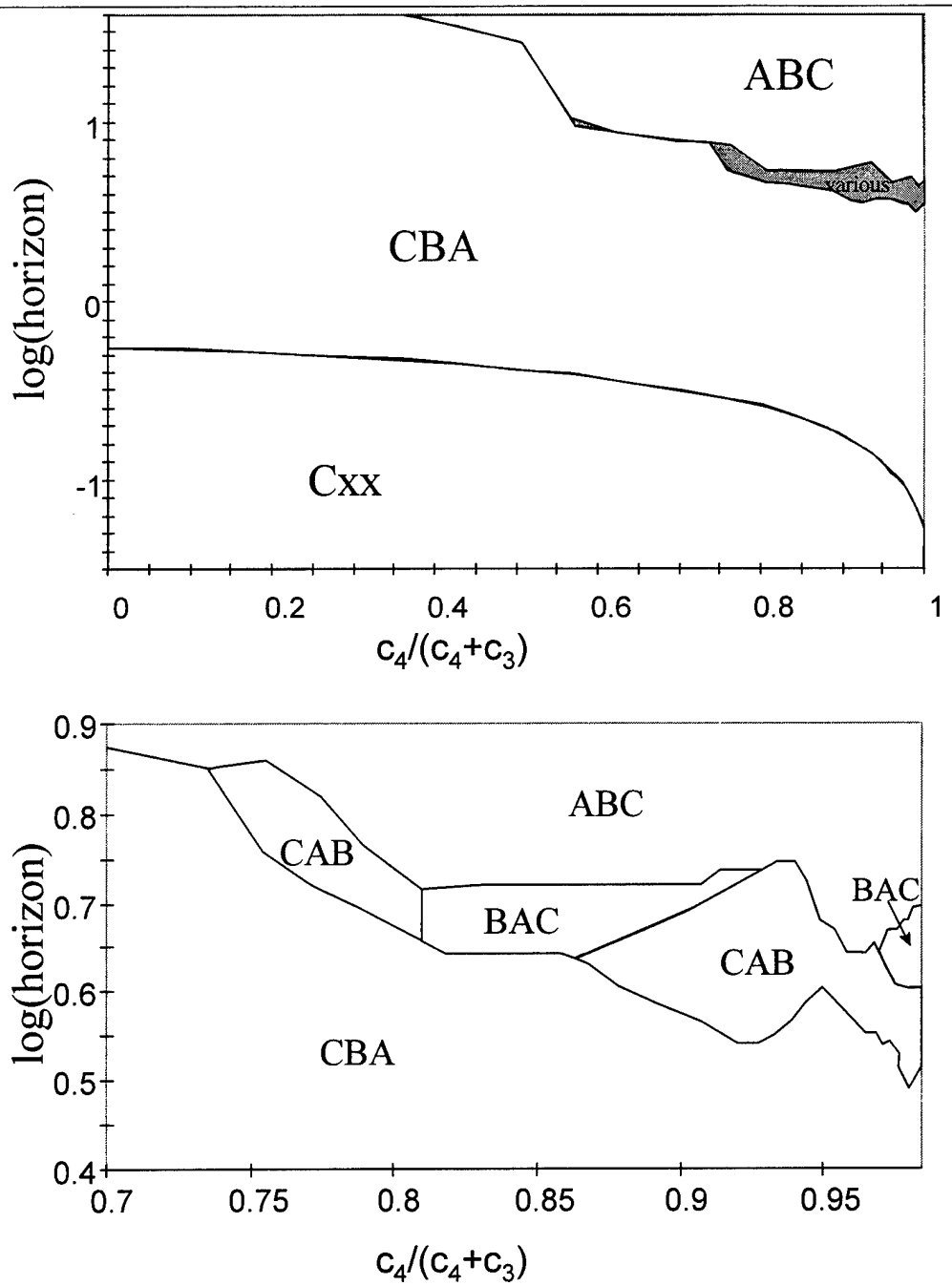


Figure 9. Plot of optimal sequences. Optimal sequences plotted for a 3-customer example in which the horizon and overtime unit cost are varied. The region marked "various" in the upper plot contains a number of different optimal sequences. This region is shown in more detail in the bottom plot.

his cost formulation. However, if $\tau_v > 0$, there are cases in which customers are not optimally ordered by service rate. For example, if there are three customers with exponential services, $\mu_A = 100.0$, $\mu_B = 10.0$, and $\mu_C = 1.0$, show probabilities are all 1.0, $\tau_h = \tau_v = 19$, $c_2 = c_3$, and $c_4 = 100c_3$, then the optimal cost for arrival sequence ABC is nearly five times that of sequence ACB.

5.3 Mean Residual Life Approach

Use of a mean residual life approximation to waiting time may lead to better intuitive understanding of the strange optimal sequences obtained in some situations. Consider the example in Table 8 for the specific case of $\tau_h = 0.7$. The optimal schedules for two reasonable candidate sequences for optimum are:

$$\text{BBAA} \quad \begin{bmatrix} 0.00 & 0.47 & 0.70 & 0.70 \end{bmatrix} \quad \text{cost} = 7.601$$

$$\text{AABB} \quad \begin{bmatrix} 0.00 & 0.50 & 0.70 & 0.70 \end{bmatrix} \quad \text{cost} = 7.611$$

These schedules are close enough that one may neglect the difference in arrival time of customer 2 for the purpose of sequencing.

The choice of class for the third and fourth customers has little impact on optimal cost, since these customers are fixed at the last slot, the last slot is the selected onset of overtime, and the classes have the same means. It is reasonable that the choice of class for the second customer will have the greatest impact on cost, since its arrival time is close to those of two actual and one fictitious customers. In fact, about two-thirds of the waiting time for customer 3 is contributed solely by customer 2, regardless of the class of the first customer. The mean residual life function, $L(t) = E[\chi - t | \chi \geq t]$, has a close relationship with waiting time; the expected waiting time of customer 3 contributed solely by customer 2 can be approximated by $L(\tau_3 - \tau_2)(1 - F(\tau_3 - \tau_2))$. The first plot in Figure 10 depicts the CDFs for the services of classes A and B, approximated using a portion of the

evaluation routine in Section H.1, and the data for the other plots are approximated from these CDFs. Since in this example $\tau_3 - \tau_2 \approx 0.25$, it can be seen from the last plot in the figure that $L_A(\tau_3 - \tau_2)(1 - F_A(\tau_3 - \tau_2)) > L_B(\tau_3 - \tau_2)(1 - F_B(\tau_3 - \tau_2))$, implying that the waiting time contribution of customer 2 to subsequent customers will be greater if the second customer is of class A than if it is of class B.

The selection of the first customer's class can be made with a similar argument. Here, $L(\tau_2 - \tau_1)(1 - F(\tau_2 - \tau_1))$ is precisely $E(W_2)$. Since $\tau_2 - \tau_1 \approx 0.45$, the last plot in Figure 10 shows that $L_A(\tau_2 - \tau_1)(1 - F_A(\tau_2 - \tau_1)) > L_B(\tau_2 - \tau_1)(1 - F_B(\tau_2 - \tau_1))$ again. The effect of the first customer on the waiting times of the third, fourth and (fictitious) fifth customers has the opposite trend, but is overshadowed by the effect of customer 2. Hence the first customer should be of class B.

5.4 Local Search Approach

A number of attempted analytic approaches to the stochastic problem met with failure. The mean residual life approach just presented is at best a way of approximating waiting time. Attempts at a standard swapping approach failed because it led to unmanageable expressions, even when the customers were adjacent. Because of the inherent complexity of the problem, these analytic attempts were abandoned in favor of heuristics. As will be seen, a simple heuristic can obtain the global optimum in a majority of problems and achieve what will often be acceptable suboptimal solutions in the remaining cases. Such heuristics are of value for two principal reasons. First, there are no successful approaches to sequencing appointments at all right now, so any method is an improvement. Second, it is probable that the problem is strongly NP-hard. If that is so, any analytical approach likely would be computationally oppressive on reasonably-sized problems, while the heuristic below is polynomial.

Several local search algorithms were tested on 4- and 6-customer sequencing problems. One entailed performing the best swap of adjacent pairs at each iteration.

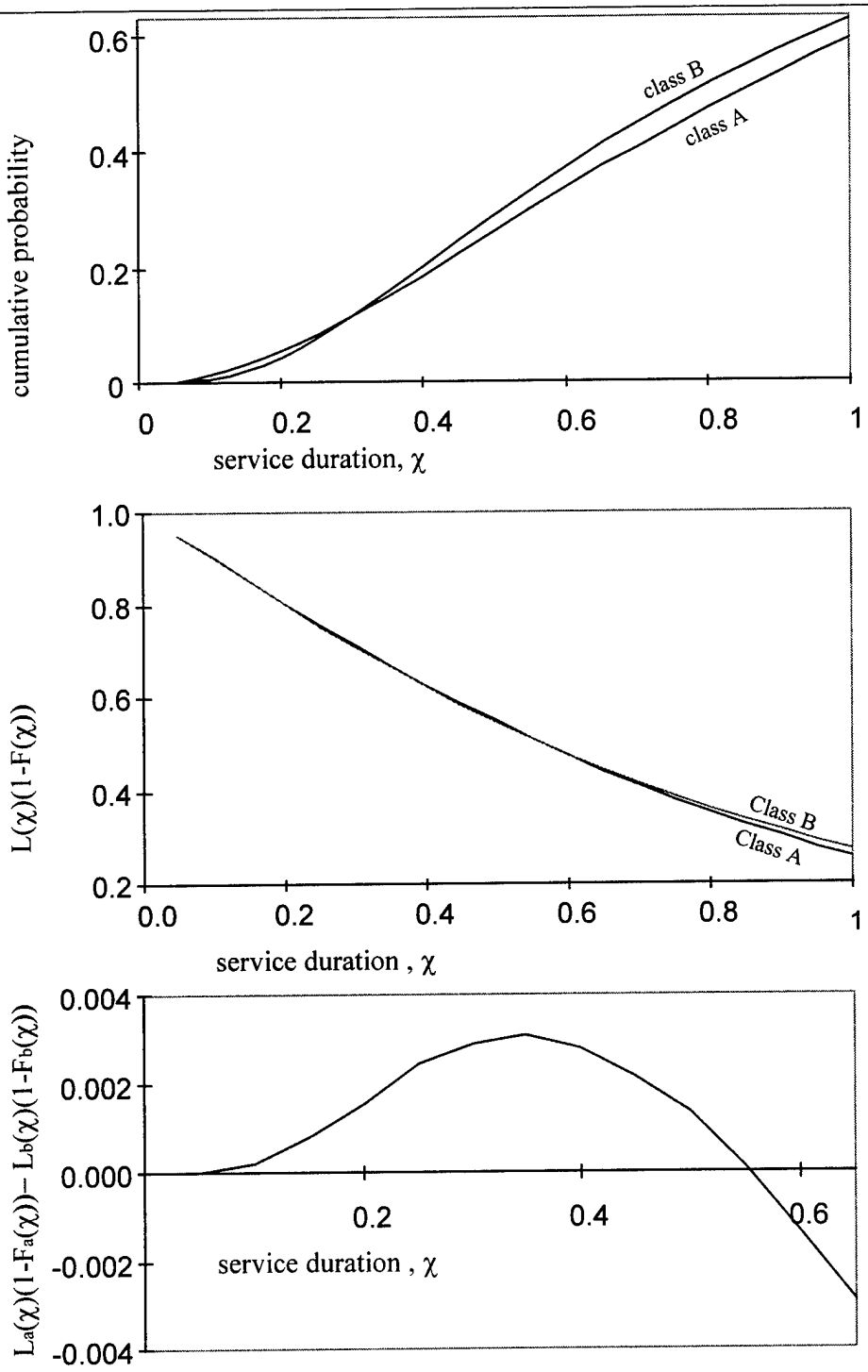


Figure 10. Comparison of three distribution measures for classes A and B in the example above. $L(\chi)(1 - F(\chi))$ is a myopic measure of waiting time.

Another entailed performing iterations that consisted of inserting each customer in turn at the spot that resulted in the lowest possible cost. These candidates were suggested by Pinedo [130: p148]. The best performance by far, both in terms of speed and accuracy, was obtained by the following algorithm:

Sequencing Algorithm

1. Select an initial sequence, Π. Determine the cost of the optimal schedule for Π.
2. Perform each of the $\binom{N}{2}$ possible pairwise swaps on Π, and determine the cost of the optimal schedule for each resulting sequence.
3. If the best swap in step 2 improved on Π, replace Π and go to step 2. Otherwise, accept Π as the optimal sequence.

On its face, this is a poor approach to sequencing. Consider a sequencing problem with no inherent structure; each of the $(N!)!$ orderings by cost of the possible sequences are equally likely. For $N = 3$, it can be shown analytically that the algorithm is expected to find the global optimum 11/12 of the time. The other 1/12 of the time, the initial choice of Π is a local optimum, and the algorithm immediately fails to make progress. For other problem sizes, the likelihood of finding the global optimum for an unstructured problem can be estimated using a Monte Carlo approach. Table 11 shows that the algorithm should be spectacularly unsuccessful even on small problems, unless there is some underlying regularity to the cost of the sequences that favors the sequencing algorithm. Further, the algorithm fails regularly on deterministic problems, often obtaining costs over 50% higher than the actual optimum. Nevertheless, the next two sections quantify the success of this algorithm on a set of randomly generated stochastic problems, and Appendix E shows its effectiveness on an actual problem.

Table 11. Success rate for the sequencing algorithm on unstructured problems. Probability of the sequencing algorithm successfully finding the optimal sequence for randomly generated problems. Each of the possible (N!)! orderings of the sequences by cost is equally likely. The average number of iterations (passes through step 2 of the algorithm) is also tabulated. All but $N = 3$ are based on 10,000 runs of a Monte Carlo simulation.

N	success rate	average # passes
3	11/12	2.10
4	0.528	2.30
5	0.192	2.46
6	0.048	2.90

5.5 Experiment Design

To test the sequencing algorithm, a number of test problems were selected across the range of parameters that are expected to be encountered in scheduling/sequencing problems. For each problem, the global optimum was approximated using the algorithm and actually determined by optimizing the schedule for each possible sequence. Coxian parameters were determined using the methods developed in Appendix F. The following ranges and characteristics were selected.

- Number of customers. Because exhaustive enumeration is required to check the algorithm, the time required to test a problem becomes prohibitively long for even small values of N. For $N = 6$, for example, a single design point requires 15 minutes on a 133 MHz Pentium. For $N = 10$, the run time is estimated at over 50 days. From Table 11, the algorithm for general problems should only be successful about half the time for $N = 4$ and about 5% of the time for $N = 6$. Any effectiveness on this problem should show up with these values of N if it exists and should carry over to larger values of N.
- Number of schedule slots. From preliminary tests, it seems that the value of K has no measurable effect on the efficacy of the algorithm. It was held fixed at 11 for the 4-customer tests and at 21 for the 6-customer tests. These are realistic values.

- Overtime point, τ_v. The overtime point was not considered a large factor in the effectiveness of the algorithm and was fixed at the schedule horizon, which is realistic for many problems.

- Service distribution means. The mean of each customer's service was selected independently from a lognormal distribution. The log of the service mean had a mean of zero and a variance of 0.1, 1.0, or 10.0. The variance of the logs of the means will be denoted as VARMEAN in the following discussion. These values were selected in order to test both cases in which means for a series of customers were closely clustered and those in which some means were outliers.

- Number of phases. The service distributions were limited to those represented by at most 4 Coxian phases. Figure 32 (with $r = 2$) shows that this is a reasonable compromise between the goals of maximizing the reachable 3-moment space and keeping computation time reasonably small.

- Service coefficients of variance. Each c was selected independently from the set [0.5, 0.8, 1.0, 1.2, and 1.5]. The low value is limited by the choice of only four phases. The high value is an extreme for realistic problems.

- Third service moment. Other research indicated that when $c \leq 1$, the third moment was not critical to the cost for similar queueing problems [3], and preliminary research indicated no dependence of optimal sequence on third moment in that region. For those reasons, the third moment was left uncontrolled when $c \leq 1$. When c was chosen as 1.2, the third noncentral moment was chosen as either 6.05 times the cube of the first moment or 30.0 times the cube of the first moment, with equal probability. When c was chosen as 1.5, the third noncentral moment was chosen as either 7.8 times the cube of the first moment or 30.0 times the cube of the first moment. The low choices represent the lowest possible values, given four Coxian phases (*cf.* Section F). The high choices represent reasonably high values.

- Customer unit costs. These were chosen independently from a lognormal distribution, with the log of the cost having a mean of zero and variance of 0.5. This ensured a mix of problems with close unit costs with some that had outliers.
- Starting point. For each trial, the algorithm was started from one of three sequences (noted by START in the following discussion):
  - Selected at random
  - Ordered by weighted shortest processing time (WSEPT)
  - Ordered by weighted variance of the processing time (WSVPT)

For each of the parameter sets chosen from the above considerations, a set of 100 experiment design points were selected by varying the server overtime unit cost and the schedule horizon regularly. The overtime unit cost was tested at the values [0.1, 1.0, 10.0, 100.0, 1000.0]. For each of these values, τ_h was tested at 20 points spaced uniformly between 0.1 and 2.0. These two parameters were selected for more regular testing because preliminary experiments indicated they had a strong effect on the optimal sequence.

5.6 Experiment Results

The results of 10,000 4-customer trials are summarized in Table 12. To check the relative effectiveness of the algorithm over different groupings of customer means, the variance of the logs of the means were generated randomly in each trial from one of three standardized lognormal populations, as discussed in the preceding section: 4200 trials at VARMEAN= 0.1, 2477 trials at VARMEAN=1.0, and 3323 trials at VARMEAN=10.0. The efficacy of the algorithm was measured in terms of:

- The percentage of time the algorithm returned a sequence whose cost was the global optimum or was within 0.001% of the cost of the global optimum (% success). This acceptance of near-optima avoids potential problems with small floating-point errors in the cost computation routines.

- The average percent error of the conjectured optimal cost from the actual optimal cost (MAPE).
- The average error divided by the average cost (% error). This measure avoids heavily weighting percent errors that are high due to the cost being quite low.
- The maximum percent error of the conjectured optimal cost from the actual optimal cost (max % error).
- The average number of iterations (passes through step 2 of the algorithm) required to reach the conjectured optimal cost (ave iter).
- The maximum number of iterations required to reach the conjectured optimal cost (max iter).

Table 12. Four-customer experiment results

VAR-MEAN	START	% success	MAPE	% error	max % error	ave iter	max iter
0.1	WSEPT	96.1%	0.076%	0.0023%	8.4%	2.45	9
0.1	WSVPT	95.5%	0.092%	0.0016%	15.6%	2.71	9
0.1	random	95.8%	0.081%	0.0019%	12.7%	3.79	10
1.0	WSEPT	91.8%	0.080%	0.0066%	14.2%	2.07	7
1.0	WSVPT	91.9%	0.109%	0.0050%	23.9%	1.82	8
1.0	random	91.2%	0.131%	0.0080%	16.5%	4.11	11
10.0	WSEPT	98.9%	0.009%	$3 \cdot 10^{-7}$%	2.6%	1.34	5
10.0	WSVPT	98.6%	0.010%	0.0011%	2.0%	1.29	4
10.0	random	82.1%	7.790%	0.1754%	638.8%	4.00	13

Goodness-of-fit tests (Chi-square and Kolmogorov-Smirnoff) suggest that the observed errors and percent errors obtained from each starting point are exponentially distributed, as are the numbers of iterations required.

There were a number of spectacular failures at VARMEAN=10.0 for starting sequences chosen at random. Because a random starting point is ineffective for some problems, it was not utilized in subsequent trials. While the results starting at WSEPT and WSVPT appear quite similar, WSEPT yields a smaller mean average percent error in each set of trials. Among the cases in which the candidate optima

found by starting at WSEPT and WSVPT differed, 46% of the WSEPT starts were better than the WSVPT starts at VARMEAN=1.0, compared to 100% for VARMEAN=10.0 and 74% for VARMEAN=0.1. This would lead one to prefer a WSEPT start, and this was the course taken in the remainder of this effort.

For each of the above sets of trials, the magnitude of errors appeared to be related to the size of the overtime cost coefficient. When the domain was restricted to those trials that did not locate the optimum, the correlations between the percent error and $\log(c_5/\sum_{i=1}^{4} c_i)$ fell between -0.37 and -0.57. A larger relative value of the overtime cost coefficient seems to exert a stabilizing influence on the algorithm. This may have to do with the fact that the larger coefficient leads to optimal schedules that are earlier, and thus vary less when the sequence is modified. Although decreasing τ_h also shifts the optimal schedule earlier, an opposite effect was observed; the correlation between $\tau_h/\sum_{i=1}^{4} E[\chi_i]$ and error ranged from 0.25 to 0.58.

A set of 6-customer experiments were run with WSEPT as the starting point. VARMEAN was set to 1.0, since this intermediate value generated the poorest results in the 4-customer experiment. Table 13 compares the results with the 4-customer results found above. While the algorithm clearly does not perform as well with 6 customers, the results are still acceptable for most applications.

Unfortunately, validation of the sequencing algorithm requires evaluation of the optimal schedules for each possible sequence. Because of the roughly factorial dependence of run time on number of customers, this validation is prohibitive in terms of run time for all but the smallest problems. The evaluation of the algorithm's performance for higher numbers of customers is relegated to future research.

Table 13. Comparison of four- and six-customer experiment results.

cust-omers	# trials	% success	MAPE	% error	max % error	ave iter	max iter
4	2477	91.8%	0.08%	0.0066%	14.2%	2.07	7
6	1300	85.1%	0.22%	0.0101%	13.0%	4.62	13

5.7 Summary

In summary, the optimal sequence of arrivals to an appointment system with iid stochastic services was seen to be highly dependent on a number of variables, including schedule horizon and relative sizes of unit costs. These dependencies are unpredictable without extensive calculation; the optimal arrival time of a particular customer may occur at the beginning of the sequence for one set of parameters and at the end for a slightly changed set. Further, identical customers typically are not even adjacent in the optimal sequence, a rare situation in sequencing problems. The seemingly erratic behavior of the optimal sequence is evinced even when services are deterministic.

While a number of analytical approaches to sequencing arrivals to an appointment system with iid stochastic service times were unsuccessful, an effective heuristic was determined. Tests of up to six customers succeeded in finding the global optimum over 85% of the time and maximum error over thousands of tests was 14%.

Despite the long history of appointment system analysis, the fact that certain sequences resulted in lower appointment system costs was not recognized in the literature until this decade [164]. Up to now, optimal sequencing has addressed extremely limited cases, and attempts to solve those have been unsuccessful [162, 164]. This dissertation is the first recognition of the odd behavior of the optimal sequences, the first observation that the optimal sequencing problem has its roots in the deterministic analogue, and the first successful solution approach to the sequence optimization of stochastic service systems.

VI. Conclusion

This chapter summarizes the research performed in this dissertation, including the appendices following. It highlights its unique contributions and proposes directions for future research.

6.1 Contributions

This research has contributed materially in a number of ways to the goal of optimizing arrival times to an appointment system. First, the cost function used is much broader than those used in the past. It incorporates the effects of no-shows and lateness. No past effort has considered the effects of no-shows. Lateness was only incorporated in cost evaluation in the case of identical service distributions and identical interarrival times. The cost function used here employs a generalization of overtime that unifies the individual measures of server availability, idle time, and overtime used in other works.

Other efforts have employed an embedded Markov chain approach to appointment system cost evaluation, as this work did. By doing so, distinct phase-type service distributions are admitted, which can approximate general service distributions to arbitrary accuracy. However, those efforts relied inherently on Jordan decomposition for matrix exponentiation, a procedure that was shown here to err substantially in floating-point implementations when eigenvalues are nearly confluent. No such difficulty was encountered in this approach.

Second, an approach to optimization of the schedule of arrivals over a lattice in this work was developed. This method relied on the piecewise convexity and submodularity of the cost function when lateness is not allowed. The only other efforts that addressed optimization of the scheduled arrival times over a lattice were limited to identical Erlang service times. Further, these methods were shown to be less efficient than the one proposed.

Third, optimal sequencing of appointments has only been considered for systems with two customers or for systems with exponential services and equal unit waiting costs. Here, a heuristic approach to sequencing was introduced that admits arbitrarily accurate approximations to distinct general distributions, distinct no-shows, and arbitrary unit costs.

Last, a new approach to matching moments using Coxian distributions was introduced. It was proven that a Coxian phase appended to an Erlang distribution can match the first three moments of any given distribution with positive support, requiring only four parameters to be determined. This parsimonious representation could be of use in numerous other models that employ phase-type distributions.

6.2 Future Research

There is a great deal more research to be done in the area of appointment system optimization. The proposed approach to sequencing is heuristic and does not always lead to the global optimum; perhaps a better heuristic approach might be found. Also, the proposed solution to the problem of optimizing the set of combinations can be improved and requires further attention before actual implementation. In particular, it assumes the same number of customers are scheduled each day. If this restriction were relaxed, some addition to the cost function would be necessary to account for the value of service, which previously was a constant.

While the appointment system model considered here is more general than those considered in other research, there are still assumptions that are unrealistic and that future research should attempt to remove. For instance, a scheme was developed for approximating schedule cost if customer lateness is allowed. However, modeling lateness destroys convexity and submodularity, both of which were essential to the arguments presented in Chapter IV, so this feature was not incorporated into the optimization algorithms. Very little has been done in the past even in terms of evaluating expected waiting time under lateness, and no research has been

done to show what the effects of lateness are on the optimal sequence, schedule, or cost. These are important questions that this dissertation may provide an initial framework for addressing.

A single-server queue was assumed. If the server is replaced by a network of servers, submodularity still holds (by the same proof as that of Theorem 2), so the lattice scheduling algorithms still are effective. A cost algorithm for this situation can be generated by expanding the state space of the continuous imbedded Markov chain. It is unclear how effective the sequencing algorithm would be.

Other features, such as balking, preemption by unscheduled customers, server failures, rescheduling of no-shows, and dependence of service time on factors such as current queue length, are important considerations in some applications. However, they have not been incorporated into the current model.

It may be possible to expand the model to consider more general cost functions as well. Currently, the cost is a linear combination of server overtime and each customer waiting time. One might add another overtime term with a new overtime point in order to increase the cost of overtime beyond a certain time.

The heuristic sequencing algorithm might be more effective if incorporated into a general search algorithm, such as a Tabu search. This could allow rapid location and rejection of local minima. Wang first proposed such an approach in a recent conversation with the author.

The deterministic sequencing problem was formulated but not solved. While this does not seem to be a good model of any practical situation, its solution might lead to some insight into the stochastic sequencing problem. In particular, it might help explain why the sequencing algorithm works well on stochastic problems but often performs poorly on deterministic ones. This could lead to prediction of which stochastic problems the sequencing algorithm will fail to solve satisfactorily.

A reverse course of inquiry may be profitable as well. Could it be that transforming a deterministic scheduling problem into a stochastic one would improve the performance of the sequencing algorithm and yet allow the optimal sequence of the deterministic problem to be determined? Such "stochastization" approaches have proved helpful on problems such as determining the three-dimensional geometry of a set of objects, given distance information [110]. The main advantage to such an approach is a smoothing of local optima, which is precisely the problem encountered in the deterministic sequencing problem. While the solution of the deterministic sequencing problem is not of tremendous import, it could lead to effective solution methods of its close relative, the $1||\sum w_j T_j$ problem.

As noted here and by Topkis, there is a wealth of functions whose submodular structure can be exploited [155]. Application of the fixed-lattice algorithm to other submodular lattice problems might be more effective than current approaches.

However, as the preliminary study in Appendix E made clear, no further analytical research need be done in order to use this work to achieve vast improvements over the current state of appointment systems. Currently, no practitioner has attended to the question of optimal sequence of arrivals, and few pay attention to any quantity but the service mean when determining a schedule of arrivals. Every appointment system in which the waiting time of customers has value, whether it be patients arriving to a doctor's office, parts arriving to a just-in-time system, cargo planes arriving to an unloading facility, or fighter planes arriving to a practice range, can benefit from the approaches offered in this dissertation. Of all the contributions future researchers may make, possibly one of the most important is the practical demonstration of the efficacy of this work.

6.3 Summary

This concludes one researcher's attempts both to solve the 46-year-old problem of optimally scheduling and sequencing arrivals to an appointment system and to

extend the current set of tools available to the analyst. The first goal - that of establishing effective approaches to the optimization problems – was achieved. For the first time, a heuristic algorithm has been presented that approximates the optimal customer arrival time sequence, and an effective new analytic algorithm has been presented for scheduling the arrival times of these customers. This has opened the way for other researchers to pursue higher-level problems, such as the optimization of customer combinations.

The second goal of developing new analytical tools also was achieved. The scheduling algorithm has been shown to apply to the optimization of submodular functions, a class with numerous practical applications. Common approaches to matrix exponentiation have been examined for numerical stability and some new results obtained. A parsimonious approach to selecting a Coxian distribution with a given set of first three moments has been presented.

The issue of appointment system optimization is far from closed. The previous section highlighted a number of important problems that have not yet been addressed. It is hoped that this research on optimizing appointment systems will lay a foundation for future work and that the tools developed here will prove of use to other researchers.

Appendix A. Deterministic Analogue

Important insights into the nature of the scheduling and sequencing of arrivals to an appointment system can be gained by considering a simple deterministic appointment system analogue. These insights are integral to the arguments in Chapter V, but the supporting arguments are contained here to improve readability. This appendix will prove the optimal schedule for a given sequence of arrivals, given deterministic services. The determination of the optimal sequence will be shown to be a nonlinear knapsack problem, and thus probably NP-hard. Some special cases are proved to be easily solvable.

Let service times be deterministic and known. No-shows are not allowed, and customers are punctual. N customers are to be assigned arrival times between 0 and τ_h (as well as an arrival sequence) such that the weighted sum of the customer waiting times is minimized:

$$\text{Minimize: } C(\tau) = \sum_{i=1}^{N} c_i W_i(\tau) \quad \text{such that } \tau_i \in [0, \tau_h] \ \ \forall i \tag{30}$$

The overtime term is omitted, which simplifies the following treatment while retaining the salient features of the solution. For simplicity, it will also be assumed that arrival times are restricted to a lattice with regular intervals of size Δ and that customers service times are strictly positive integral multiples of Δ.

This deterministic appointment system is not very realistic. In most conceivable circumstances, the horizon would be modified to equal the sum of service times, in which case optimal schedules and sequences can be obtained trivially. However, in the context of the larger stochastic problem, in which the service time sum is not known in advance, this system is quite relevant. Most importantly, the seemingly chaotic nature of the optimal sequence can be seen to have its roots in this simple

problem. In the following treatment, in order to avoid the trivial solution, assume the horizon is less than the service sum, forcing some customers to wait for service.

For a fixed sequence, the optimal schedule of arrivals can be constructed easily. This will be proved recursively with the help of two propositions.

Theorem 13 *In the deterministic problem with $\tau_h = 0$, any optimal sequence is WSPT.*

Proof: Since $\tau_h = 0$, then the schedule is fixed at $[0\ 0 \cdots 0]$, and $W_j = \sum_{i=1}^{j-1} \chi_i$. This sequencing problem is one of minimizing weighted completion time under simultaneous arrival times, which can be shown by a simple swapping argument to be WSPT [130: Theorem 3.11]. ▪

Lemma 14 *Given deterministic arrival times and $\tau_h \leq \chi_1$, the optimal schedule for a given sequence places the first customer at 0, and the rest at τ_h.*

Proof: From the nontriviality condition above, at least one customer has a nonzero waiting time. The optimal schedule has no idle time; if it did, then all subsequent arrival times could be moved earlier, and the time gained could be used to extend the interarrival time prior to the next customer. By reference to Figure 1, a simple bookkeeping argument shows that

$$W_j(\tau) = \max\left[0, \sum_{i=1}^{j-1} \chi_i - \tau_j + \tau_1\right] \quad \forall j \in [2, N] \tag{31}$$

It is not possible to change the summation, since the order is fixed. Each W_j is minimized by minimizing τ_1 – *i.e.*, setting it to zero. Each W_j is linear in τ_j for $\tau_h \leq \tau_1$, so W_j is minimized by maximizing τ_j – *i.e.*, setting it to τ_h. Thus, the lowest cost is achieved when the first customer arrives at zero and all others at τ_h. ▪

Lemma 15 *For a fixed sequence and deterministic arrival times, the optimal cost for a problem with $\tau_h \leq \chi_1$ is identical to the one in which $\tau_h = 0$ and the service of the first customer is reduced by τ_h.*

Proof: Since $\tau_h \leq \chi_1$, each customer besides the first is optimally scheduled at τ_h, and a reduction of τ_h results in a commensurate reduction of each customer's arrival time except the first. By inspection of Equation (31), it is clear that the same $W_j(\tau)$ is obtained when both τ_j and χ_1 are reduced the same amount, so the optimal cost is unchanged. ▪

Theorem 16 *In a deterministic problem, the optimal schedule for a given sequence places each customer at*

$$\tau_j = \min \left[\tau_h, \sum_{i=1}^{j-1} \chi_i \right] \tag{32}$$

Proof: Consider the following recursive procedure, initializing $N_h = 1$: Set $\tau_h = 0$, so that each arrival time is at zero. Increase τ_h by either the service time of the initial customer in this transformed problem or by the amount needed to reach the desired value of τ_h, whichever is smaller. In the former case, the first interarrival time is equal to the first service time, so the second customer experiences no waiting time and no idle time. The optimal sequence schedules a single customer at zero and the remainder at τ_h. If τ_h equals its desired value, stop. Otherwise, use Lemma 15 to transform the problem again by reducing χ_1 and τ_h to zero, then increment N_h and repeat the procedure.

When the procedure is completed (guaranteed in at most $N - 1$ iterations), each interarrival time between customers i and $i + 1$ is equal to χ_i, when $i < N_h$. Each interarrival time when $i \geq N_h$ is zero, so the proof is complete. ▪

Now that the optimal schedule for a given sequence is proved, the optimal sequence can be considered. Theorem 13 proved the optimal sequence for $\tau_h = 0$. Now consider increasing τ_h:

Theorem 17 *In the deterministic problem with* $\tau_h \leq \max[\chi_1, \chi_2, \ldots, \chi_N]$, *the optimal sequence is one in which the first customer has smallest* $(\tau_j - \tau_h)/c_j$, *and the rest are WSPT.*

Proof: Let ξ be the current value of the horizon, for convenience in the subsequent transformations. By Theorem 16, the optimal schedule will have one customer at zero and the rest at ξ. By Lemma 15, the cost for a particular sequence is equivalent to that obtained when $\tau_h = 0$ and the first customer's service is reduced by ξ. Assuming the first customer in the optimal sequence can be determined, this transforms the problem to one with $\tau_h = 0$, and the optimal sequence is WSPT by Theorem 13.

The problem now is to prove which should be the first customer, the one to be transformed by reduction of service time. Consider the current transformed problem with τ_h, and suppose all service times were reduced by ξ, rather than just the first. By Theorem 13, the first in the optimal schedule of this new problem would be the one with lowest $(\tau_j - \xi)/c_j$. Relaxing the service times of each but the first customer from $\chi_j - \xi$ back to χ_j might change the order of subsequent customers, but the first customer in the optimal schedule would remain the same. Thus, the optimal sequence is one in which the first customer is the one with lowest $(\tau_j - \xi)/c_j$ and scheduled at zero, with subsequent customers in WSPT order and scheduled at ξ. ∎

The previous propositions show that, when τ_h is smaller than the smallest service, the optimal schedule and sequence can be determined rapidly. But when τ_h is larger, it is not clear how many customers are to be scheduled before τ_h, complicating things considerably. Let N_h be the index of the first customer optimally scheduled at τ_h for a given sequence. Let $d_i = 1$ if customer i is to be scheduled before τ_h and zero otherwise. Then, assuming the optimal customer sequence can be found for a given d, the goal is to minimize $C(\tau)$ subject to $\sum_{i=1}^{N} d_i \chi_i \leq \tau_h$.

Now consider the question of the optimal customer sequence given a vector d. A given d divides the customers into those arriving prior to τ_h and those arriving at

τ_h, and both sets require optimal ordering. From the previous arguments,

$$W_j(\tau) = \begin{cases} 0 & \forall j \leq N_1 \\ \sum_{i=1}^{j-1} \chi_i - \tau_h & \forall j > N_1 \end{cases} \tag{33}$$

$$C(\tau) = \sum_{j=N_1+1}^{N} \left[\sum_{i=1}^{j-1} \chi_i - \tau_h \right]$$

The same optimal schedule cost will be obtained for every sequence in which the same set of customers is scheduled to arrive before τ_h, regardless of the order of this set. For the set of customers arriving at τ_h, the same argument as that used in Theorem 16 can be used to show that the $(N_H)^{st}$ customer must be the one from this set with smallest $\left(\chi_j - \sum_{i=1}^{N_H-1} \chi_i - \tau_h\right)/c_j$. Thus, once d is selected, the sequence is determined. The number of possible sequences to be considered is thereby reduced from $N!$ to 2^N. Now the problem is reduced to:

$$\begin{aligned} &\text{minimize:} \quad C(\tau) = \textstyle\sum_{j=1}^{N} d_j \max\left[0, \sum_{i=1}^{j-1} \chi_i - \tau_h\right] \\ &\text{subject to:} \quad \textstyle\sum_{i=1}^{N} d_i \chi_i \leq \tau_h \ \text{ and } \ d_i \in \{0,1\} \ \forall i \end{aligned} \tag{34}$$

which is a knapsack problem, albeit one with an objective that is nonlinear, both due to the max function and to the summation changing with the ordering of the customers. Thus, when d_i changes value, the contribution of a number of customers to the cost may change.

Since even the knapsack problem with linear cost function is NP-hard, it is reasonable to suspect this problem is also. Lawler addressed the problem of determining the optimal sequence that minimizes the total weighted tardiness, given due dates (the $1||\sum w_j T_j$ problem, in currently accepted sequencing theory notation), and proved it is strongly NP-hard [92]. Set all due dates to τ_h in the current problem. Since the waiting time of the j^{th} customer is equivalent to the tardiness of the

$(j-1)^{st}$ customer, this problem can be thought of as a variation on Lawler's, lending further evidence that this problem is also strongly NP-hard.[1]

However, there are situations in which the optimal sequence is quite easy to determine. As discussed above, when τ_h is smaller than the smallest service time, the optimal sequence and schedule are easily determined, in the time required to sort two sequences of length N. Another special case can be deduced by considering Equation (34). There are three ways to manipulate the arrival sequence to minimize the cost. One is to reduce the number of terms in the summation by making N_H as large as possible. Another way is to minimize the c_j for $j > N_H$. The last is to minimize the contributions of the $\sum_{i=1}^{j-1} \chi_i$ expressions. The first and last goals can be accomplished by ordering the customers by smallest to largest service. The second goal can be accomplished by ordering customers by largest to smallest unit cost. If these two orderings coincide (*i.e.*, if it is true that $\chi_i \leq \chi_j$ implies $c_i \geq c_j$ $\forall i, j$), optimality is assured. A set of customers possessing this property is said to have *agreeable* weights.[2]

The sequencing algorithm advocated in Chapter V performs poorly on the deterministic problem, and one can set up problems in which the error in the optimum found by this method is arbitrarily large. The fact that the algorithm performs so well on stochastic problems hints that some smoothing away of local optima takes place, allowing the algorithm to attain the global optimum. This suggests that the addition of a similar smoothing operation might be a profitable way of attacking the $1||\sum w_j T_j$ problem.

In conclusion, although the deterministic problem of sequencing and scheduling customers to an appointment system is not particularly realistic, it does exhibit the major features seen in stochastic problems, as was shown in the examples in Chapter

[1]Thanks to Michael Fredley, Captain, USAF, for pointing out this connection.
[2]E. L. Lawler first used *agreeable* in this way [91].

V. Even this gross simplification of an appointment system appears be NP-hard, suggesting that the stochastic sequencing problem also is NP-hard.

Appendix B. Application of the Lattice Algorithms to Other Problems

This appendix explores the implications of the convexity and submodularity of the cost function for the scheduling problem. This structure will be seen to be shared by other problems, and the proposed search algorithms may be of use with them as well.

Submodularity, together with convexity of the cost function with respect to each arrival time, imply an important structure of the Hessian matrix of the cost function (or the discrete analog to the Hessian, if the Hessian does not exist). Suppose $C(\tau)$ is twice-differentiable with respect to τ. First, since it is convex with respect to each τ_j, it must be that $\frac{\partial^2 C}{\partial \tau_j^2} \geq 0$ for all j.

The proof of Theorem 2 constructed S_2 from S_1 by shifting customer i's arrival time later by δ_i and constructed S_1' (S_2') from S_1 (S_2) by shifting customer j's arrival time later by δ_j (Figure 8). It was then proved that $[C(S_2') - C(S_2)] - [C(S_1') - C(S_1)] \leq 0$, which holds for all i, j and all positive values of δ_i, δ_j that do not change the order of arrivals. Then

$$\lim_{\delta_j \to 0} \left(\frac{[C(S_2') - C(S_2)] - [C(S_1') - C(S_1)]}{\delta_j} \right) = \left. \frac{\partial C(\tau)}{\partial \tau_j} \right|_{S_2} - \left. \frac{\partial C(\tau)}{\partial \tau_j} \right|_{S_1} \leq 0 \qquad (35)$$

which in turn implies that

$$\lim_{\delta_i \to 0} \left(\frac{\left. \frac{\partial C(\tau)}{\partial \tau_j} \right|_{S_2} - \left. \frac{\partial C(\tau)}{\partial \tau_j} \right|_{S_1}}{\delta_i} \right) = \left. \frac{\partial^2 C(\tau)}{\partial \tau_j \partial \tau_i} \right|_{S_1} \leq 0 \qquad (36)$$

Thus, if the Hessian exists, it has a nonnegative diagonal, with nonpositive entries at all other positions. This Hessian structure of submodular functions was first noticed by Lorentz [101].

The search algorithms presented in this chapter are based on the cost function being submodular as well as convex, both with respect to some vector that associated a scalar with each customer. The optimal cost with respect to this vector is then sought. The lattice algorithms proved effective for the scheduling of arrivals problem, and it is logical to seek other problem classes for which the algorithms might be useful.

Topkis pointed out a number of submodular problems, including the max-flow, min-cut problem, an optimal advertising strategy problem, and optimal control of an unreliable system [155]. Other more mathematical problems have been explored as well [32, 37, 101, 103]. Another class of problems is proposed here.

One example of this class considers optimization of spatial position rather than time. Consider a collection of point charges, each pair of which repels each other with a force proportional to the inverse square of the distance between them and to the magnitudes of the two charges. If the first and last point charges are fixed, and the others are constrained to move on the line segment between them, does an equilibrium exist? If so, what is the equilibrium position of the charges? Is it unique? Here, the goal would be to determine the position vector(s) for which the net force on each charge is zero. Another equivalent and more pertinent approach would be to equate cost with the total potential energy of the system and minimize it, where the potential energy contribution of each pair of charges is proportional to the inverse of their separation and to their magnitudes. This cost function is submodular and convex, given a particular ordering of charges on the line segment is maintained. While such a problem might not require a lattice solution, it has been seen that the variable-lattice algorithm is superior to typical NLP methods for rough approximations of the optimum.

Cost functions in which entities tend to "repel" each other within a constrained area are here termed *jostling* functions. Such problems arise frequently in physics. The optimal locations of electrons around a set of nuclei is a common problem.

Other applications include resource allocation problems, such as the positioning of microwave relay towers in an area so as to optimize signal strength and minimize redundant coverage. It will be seen that jostling functions are submodular and convex over a single spatial dimension, but only certain jostling functions retain submodularity for more complex domains.

In general, if $C(\tau)$ is submodular for some function C, it is not true that $f(C(\tau))$ is also submodular. In fact, unless C always imposes certain orderings on the costs, Lair and Oxley showed that $f(x)$ must be an increasing linear function [87]. Topkis proved that if $C(\tau)$ is monotone and submodular and $f(x)$ is convex and increasing, then $f(C(\tau))$ is monotone and submodular [155]. One important special case occurs when $C(\tau)$ is separable into convex terms, each of which only depend on the difference of two arrival times: $C(\tau) = \sum_{i=1}^{N+1} \sum_{j=i}^{N+1} C_{ij}(\tau_j - \tau_i)$, where each C_{ij} is a convex function.

Theorem 18 *Let C_{ij} be convex functions for all i and j. If*

$$C(\tau) = \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} C_{ij}(\tau_j - \tau_i) \tag{37}$$

then $C(\tau)$ is submodular.

Proof: One need only consider C_{ij}, since it is the only term of C that varies with respect to both i and j. For notational convenience, then, let $C(\tau) = C_{ij}(\tau_j - \tau_i)$. Let $\lambda = \delta_i/(\delta_i + \delta_j)$, $x = \tau_j - \tau_i - \delta_i$, and $y = \tau_j - \tau_i + \delta_j$. Then

$$\begin{aligned} C(S_1') &= C_{ij}(x) \\ C(S_2) &= C_{ij}(y) \\ C(S_1) &= C_{ij}(\tau_j - \tau_i) = C_{ij}(\lambda x + (1-\lambda)y) \\ C(S_2') &= C_{ij}(\tau_j - \tau_i - \delta_i + \delta_j) = C_{ij}((1-\lambda)x + \lambda y) \end{aligned} \tag{38}$$

Convexity of C_{ij} is both necessary and sufficient for [134: Theorem 4.1]:

$$\begin{aligned} C_{ij}(\lambda x + (1-\lambda)y) &\leq \lambda C_{ij}(x) + (1-\lambda)C_{ij}(y) \\ C_{ij}((1-\lambda)x + \lambda y) &\leq (1-\lambda)C_{ij}(x) + \lambda C_{ij}(y) \end{aligned} \tag{39}$$

the sum of which is

$$C(S_1) + C(S_2') \leq C_{ij}(x) + C_{ij}(y) = C(S_1') + C(S_2) \tag{40}$$

which proves submodularity. ▪

Corollary 19 *Let C_{ij} be convex functions for all i and j. Let f_{ij} be convex, non-decreasing functions for all i and j. If $F(\tau) = \sum_{i=1}^{N} \sum_{j=i+1}^{N+1} f_{ij}\left(C_{ij}(\tau_j - \tau_i)\right)$, then $F(\tau)$ is submodular.*

Proof: Each $f_{ij}\left(C_{ij}(\tau_j - \tau_i)\right)$ is convex with respect to $(\tau_j - \tau_i)$ [134: Theorem 5.1], so Theorem 18 applies. ▪

For many multi-dimensional jostling functions, the objective function is not separable, and submodularity does not hold. For instance, consider a set of entities with cost linearly dependent on functions of the Minkowski distance between each:

$$f_{ij}(x,y) = g_{ij}\left(\sqrt[k]{\mid x_j - x_i \mid^k + \mid y_j - y_i \mid^k + \ldots}\right) \tag{41}$$

Merely by inspection, it is clear that if $\frac{\partial^2 f_{ij}(x,y)}{\partial x_j \partial y_j}$ and $\frac{\partial^2 f_{ij}(x,y)}{\partial x_j \partial y_i}$ are nonzero they will be opposite in sign. Then the off-diagonal elements of the Hessian cannot all be nonpositive if the optimization is over all x and y.

On the other hand, if for each pairwise interaction, $f_{ij}(x,y)$ can be separated into terms that each involve only the difference of two coordinates, the situation changes. For instance, consider the above case if each $g_{ij}(z) = c_{ij}z^k$, where c_{ij} are

constants.

$$f_{ij}(x,y) = c_{ij}\left(\mid x_j - x_i \mid^k + \mid y_j - y_i \mid^k + \ldots\right) \tag{42}$$

Now $\frac{\partial^2 f_{ij}(x,y)}{\partial x_j \partial y_j} = \frac{\partial^2 f_{ij}(x,y)}{\partial x_j \partial y_j} = 0$, and the desired Hessian structure holds if $\frac{\partial^2 f_{ij}(x,y)}{\partial x_j^2 \geq 0}$, $\frac{\partial^2 f_{ij}(x,y)}{\partial x_i^2 \geq 0}$, $\frac{\partial^2 f_{ij}(x,y)}{\partial y_j^2 \geq 0}$, $\frac{\partial^2 f_{ij}(x,y)}{\partial y_i^2 \geq 0}$, ... for all i and j.

Even if submodularity holds for a problem with unconstrained entities, it may not hold for constrained ones. For a jostling problem with n bounds, there will be a number of entities constrained to the boundaries, possibly confounding the Hessian structure of the problem. Suppose that in a 2-dimensional problem, the i^{th} entity were constrained by the boundary $y_i = h(x_i)$. If $\frac{\partial^2 f_{ij}(x,y)}{\partial y_j \partial y_i} \leq 0$ and $\frac{\partial^2 f_{ij}(x,y)}{\partial x_j \partial y_i} = 0$ for the unconstrained problem, then by the chain rule, it is now required that $\frac{\partial h(x_i)}{\partial x_i} \geq 0$. While this requirement sometimes may be circumvented by redefinition of variables and coordinate systems, it is nonetheless a severe restriction on the nature of the allowable boundary conditions.

The algorithm is therefore applicable only to certain jostling problems in multiple dimensions, limiting its usefulness. On the other hand, if the objective function is submodular, the algorithm holds a substantial advantage over many NLP methods. For instance, a gradient search is commonly applied to electron positioning problems, even though they are nonconvex and the nature of the surface may be poorly apprehended; optimality is far from guaranteed, even after many restarts. Convexity is not required for the fixed- or variable-lattice algorithms to function, however. They will fathom the feasible space until they reach the potentially nonconvex region and stop. The lattice algorithms are guaranteed not to bypass any optima, local or otherwise. This could prove useful in fathoming the search region for subsequent analysis.

Appendix C. Effectiveness of the Lattice Algorithms

This appendix explores the number of iterations required and completion time for the fixed- and variable-lattice algorithms. Bounds are determined, as well as the functional dependence on the input parameters.

The fixed-lattice algorithm typically starts by finding S_E, the early schedule. This is not necessary, and it may be far faster to find S_L first for a problem that is quite constrained (*i.e.*, $\tau_h < \sum_{i=1}^{N} \chi_i$). On the other hand, it is necessary to start by finding S_E when there is no schedule horizon, and more efficient to do so if the schedule horizon is very large. In the following exposition, it is assumed that the search is started with S_E.

C.1 Maximum Iterations when the Horizon is Finite

For a given problem, the number of iterations to find S_E can be divided into those that improve the schedule cost (and are thus accepted as the new S_E) and those that do not improve the cost. Call these successes and failures. Define the path of the algorithm to be the sequence of schedules evaluated to reach the optimum. The number of successes encountered is independent of the path, since for any two paths, the starting and ending schedules are identical, and each iteration shifts only one customer one slot. The first customer is fixed in all schedules, so the number of successes required to reach S_E from the starting schedule $[\ 0 \quad 0 \quad \ldots \quad 0\ ]$ is $\sum_{j=2}^{N} S_E(j)$. It is not possible to predict the value of S_E in advance, but each arrival time is bounded by the horizon, τ_h, when it exists. Assume it does, and let K be the number of possible schedule slots. Then the maximum number of successful iterations occurs when $S_E = [\ 0 \quad K-1 \quad K-1 \quad \ldots \quad K-1\ ]$, and that number is $(N-1)(K-1)+1$.

The maximum number of failures is observed when every possible advancement of an arrival time fails to improve the cost, unless that failure would stop the algo-

rithm before it reached the above worst-case S_E. This occurs when the schedules [0 1 1 ... 1], [0 2 2 ... 2], ...are in the path. Between each of these schedules, there can be at most $2N-3$ evaluations, $N-2$ of which are failures. Between [0 $K-2$ ... $K-2$] and [0 $K-1$ ... $K-1$], there are no failures possible, or the algorithm would end before reaching the worst-case S_E. The total number of failures is thus bounded by $(N-2)(K-2)$, making the total number of iterations to reach S_E bounded by $(2N-3)(K-2)+N$. An example of such a worst-case search is shown in Table 14.

Table 14. Example of worst-case search for S_E using the fixed-lattice algorithm, for 4 customers and 5 schedule slots.

[**0** **0** **0** **0**]	start
[0 0 0 1]	success
[0 0 0 2]	failure
[0 0 1 1]	success
[0 0 1 2]	failure
[**0** **1** **1** **1**]	success
[0 1 1 2]	success
[0 1 1 3]	failure
[0 1 2 2]	success
[0 1 2 3]	failure
[**0** **2** **2** **2**]	success
[0 2 2 3]	success
[0 2 2 4]	failure
[0 2 3 3]	success
[0 2 3 4]	failure
[**0** **3** **3** **3**]	success
[0 3 3 4]	success
[0 3 4 4]	success
[**0** **4** **4** **4**]	success

The starting point for the search for S_L is obtained by shifting all but the first customer of S_E one slot later if possible. For the above worst-case S_E, no shifts are possible. Any earlier choice of worst-case S_E would require at most $N-1$ further iterations to obtain S_L, but this increase would be more than offset by the decrease in

iterations required to obtain S_E. Therefore, the above is also a bound to the number of iterations required to obtain both S_E and S_L. This is a substantial improvement on Simeoni's bound of $2(K-1)^2(N-1)$ [145] for the same algorithm. In the case of the variable-lattice algorithm, a bound on the number of evaluations required for each subsequent determination of S_L or S_E can be found by a similar argument.

For the fixed-lattice algorithm or the last stage of the variable-lattice algorithm, it may be that S_E and S_L differ in the arrival times of a number of customers, necessitating the enumeration of some of the schedules between S_E and S_L. There are at most 2^{N-1} such schedules, since at most $N-1$ customers can differ, and by Theorem 8, each of these can differ by at most one slot. This is the bound Simeoni proposed [145]. However, for all values of N above some minimum,[1] 2^{N-1} is greater than the number of all feasible schedules, which limits its usefulness as a bound. The problem is that many of these 2^{N-1} enumerations are infeasible, since the customers change order. These enumerations must be subtracted from Simeoni's bound.

The only situation in which customers might change order in the enumeration phase is when two customers occupy the same slot in S_E and one customer is shifted one slot later while the subsequent one is not. Let $\nu(j)$ be the number of customers arriving in slot j. The number of feasible schedules achievable by shifting only the $\nu(j)$ arrival times in slot j by Δ is equivalent to the number of $\nu(j)$-digit binary numbers in which the bits are ordered from lowest to highest, which easily is proved inductively to be $\nu(j)+1$. In addition, S_E and S_L themselves are already evaluated, as are all the schedules that differ from S_E or S_L in only one arrival and by one slot, making the maximum number of schedules to be evaluated in the enumeration phase

$$\nu(0)\prod_{j=1}^{K-2}(\nu(j)+1)-2-2\,\mathrm{sign}(\nu(0)-1)-2\sum_{j=1}^{K-2}\mathrm{sign}(\nu(j)) \tag{43}$$

[1]This minimum is dependent on K. For example, when $K=5$, $N>11$ yields a value of 2^{N-1} greater than the total number of feasible schedules, and when $K=20$, $N>60$ gives the same result.

where sign(x) is zero if $x = 0$ and one otherwise. The $j = 0$ terms of the product and summation are evaluated separately because of the special condition that the first customer remain in the initial slot. The $K - 1$ terms have no effect, since those customers in the last possible slot cannot be shifted later.

Table 15. Example of enumeration for 4 customers, 5 schedule slots, S_E = [0 1 1 2 3], and S_L = [0 1 2 3 4]

schedule	binary analogue	evaluate?
[0 1 1 2 3]	0000	no: already evaluated
[0 1 1 2 4]	0001	no: already evaluated
[0 1 1 3 3]	0010	no: already evaluated
[0 1 1 3 4]	0011	yes
[0 1 2 2 3]	0100	no: already evaluated
[0 1 2 2 4]	0101	yes
[0 1 2 3 3]	0110	yes
[0 1 2 3 4]	0111	no: already evaluated
[0 2 1 2 3]	1000	no: infeasible
[0 2 1 2 4]	1001	no: infeasible
[0 2 1 3 3]	1010	no: infeasible
[0 2 1 3 4]	1011	no: infeasible
[0 2 2 2 3]	1100	yes
[0 2 2 2 4]	1101	no: already evaluated
[0 2 2 3 3]	1110	no: already evaluated
[0 2 2 3 4]	1111	no: already evaluated

The number of evaluations during the enumeration phase is maximal when the arrival times are most evenly distributed between the first through the $(K-1)^{st}$ slots. When N is divisible by $K - 1$, the number of evaluations in the three phases is bounded by

$$(2N-3)(K-2) + N + \left[1 + \frac{N}{K-1}\right]^{K-2} \left(\frac{N}{K-1}\right) - 2K \qquad (44)$$

and a similar expression can be obtained when N is not divisible by $K - 1$. This quantity is plotted in Figure 11.

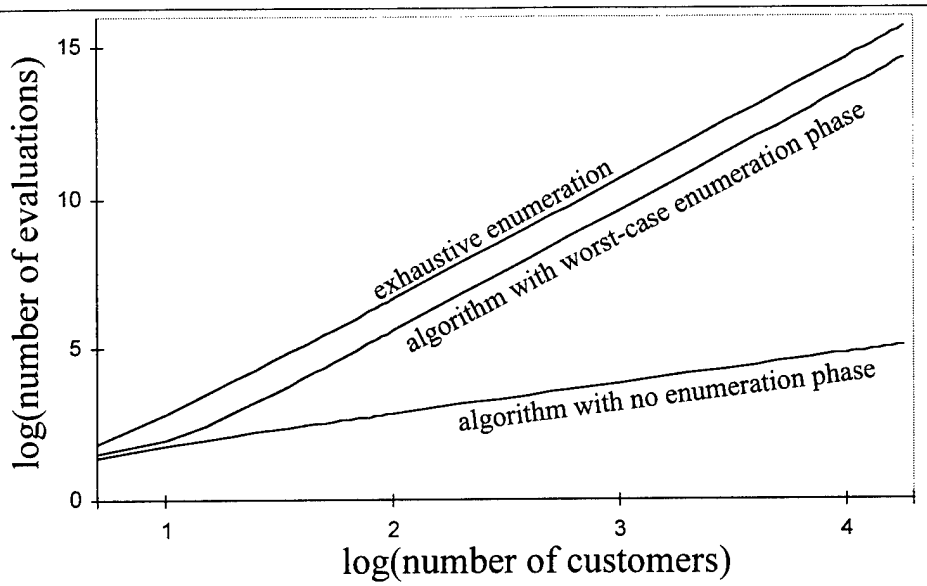


Figure 11. Dependence of the maximum number of iterations required on the number of customers for an example with 10 schedule slots

As N increases, the ratio of the maximum number of iterations required by the algorithm to the total number of feasible schedules approaches some asymptotic value. This value can be obtained analytically by using Stirling's asymptotic approximation to a factorial.

$$\lim_{N\to\infty} \frac{\left[\begin{array}{c}(2N-3)(K-2)+N+\\ \left[1+\frac{N}{K-1}\right]^{K-2}\left(\frac{N}{K-1}\right)-2K\end{array}\right]}{\binom{N+K-2}{N-1}} = \lim_{N\to\infty} \frac{\left(\frac{N}{K-1}\right)^{K-1}}{\binom{N+K-2}{N-1}}$$

$$= \lim_{N\to\infty} \frac{\left(\frac{N}{K-1}\right)^{K-1}\sqrt{2\pi}(N-1)^{N-0.5}(K-1)^{K-0.5}}{(N+K-2)^{N+K-1.5}}$$

$$= \lim_{N\to\infty} \frac{\sqrt{2\pi(K-1)}\left(\frac{N}{N-1}\right)^{K-1}}{\left(1+\frac{K-1}{N-1}\right)^{N-1}\left(1+\frac{K-1}{N-1}\right)^{K-0.5}} = \frac{\sqrt{2\pi(K-1)}}{\exp(K-1)} \qquad (45)$$

This limiting ratio is about 0.092 for $K = 5$ and 0.00093 for $K = 10$, and is reached from below, implying that even in the worst case, only a fraction of the possible schedules must be evaluated. In practice, the enumeration phase almost always is

quite small; it seldom exceeds 10 even for very large N, so the expected number of iterations is represented better by the lower line in Figure 11. If the number of enumerations required is assumed to be bounded and small, the number of iterations is linear in both N and K.

C.2 Actual Number of Iterations

The actual number of iterations required in the enumeration phase was explored in a series of tests. For $N < 5$, enumeration is never required, since S_E and S_L can differ by at most three customers. In 77,312 trials of a 6-customer, 11-slot schedule with randomly (but realistically) chosen parameters, 86.7% required no enumeration, 9.8% required an enumeration phase of 4 iterations, and 3.5% required the maximal 18 iterations in the enumeration phase. A similar 10-customer, 21-slot series of 3000 schedule optimizations yielded the CDF seen in Figure 12. The maximal number of iterations in the enumeration phase, 492, was observed 1.1% of the time.

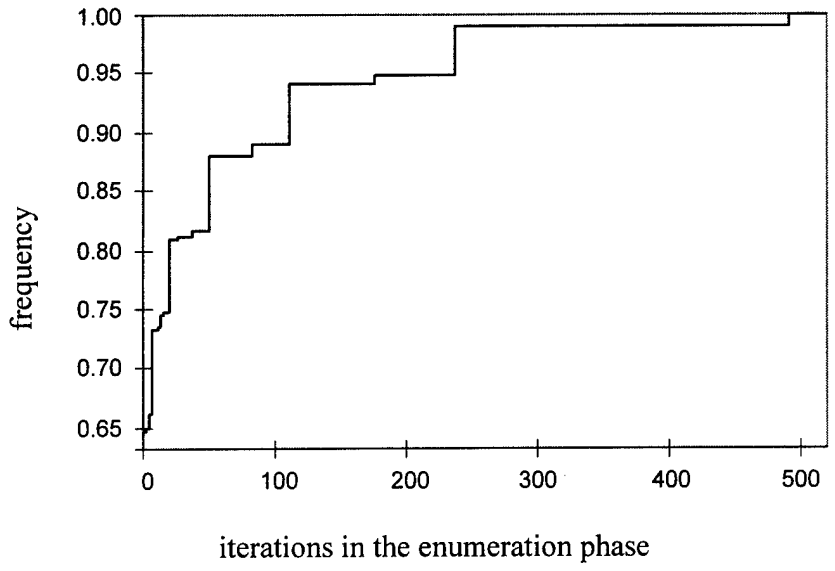


Figure 12. Estimated CDF of the number of iterations required in the enumeration phase when optimizing a 10-customer, 21-slot schedule.

The linearity of the theoretical dependence of the maximum number of iterations on the number of possible schedule slots is matched by empirical results for the typical actual number of iterations, as seen in the example in Figure 13. Here, five customers were assigned iid Erlang-2 service with mean of 2. The number of slots was varied and evenly spaced between 0 and 5, with $\tau_v = 5$ and $c_1 = c_2 = \ldots = c_6$. The points are observed values, while the solid line is the regression line. The dotted line represents the theoretical bound on the number of iterations for the algorithm if no enumerations are required, which was found above.

The remarkably linear fit seen in Figure 13 ($r^2 = 0.99998$, slope $= 0.96$) can be understood by means of an argument similar to that used for the maximum number of iterations. By Theorem 9, the search for S_E will end at the nearest or next-to-nearest lattice point that is greater than the continuous optimum schedule for each arrival. This limits the maximum number of iterations for each lattice size to approximately the same horizon, leading to a formulation similar to that above for the number of successes. The number of failures is typically negligible compared to the number of successes. The empirical fit would be expected to be poor if the enumeration phase required a disparate number of evaluations for the 13 problems. For this example, none of the optimizations required an enumeration phase longer than 6 schedules, and the two problems with the largest lattice size required no enumeration phase. This is typical.

Likewise, the actual dependence of the number of iterations on N is highly linear, as seen in the example in Figure 11($r^2 = 0.9995$). Here, the horizon and overtime point are fixed at 7 units, all cost coefficients are equal, and customers are placed into 8 slots. No optimization required an enumeration phase of more than 3 evaluations. The solid line is a linear regression, while the dotted line represents the theoretical bound on the number of iterations for the algorithm if no enumerations are required.

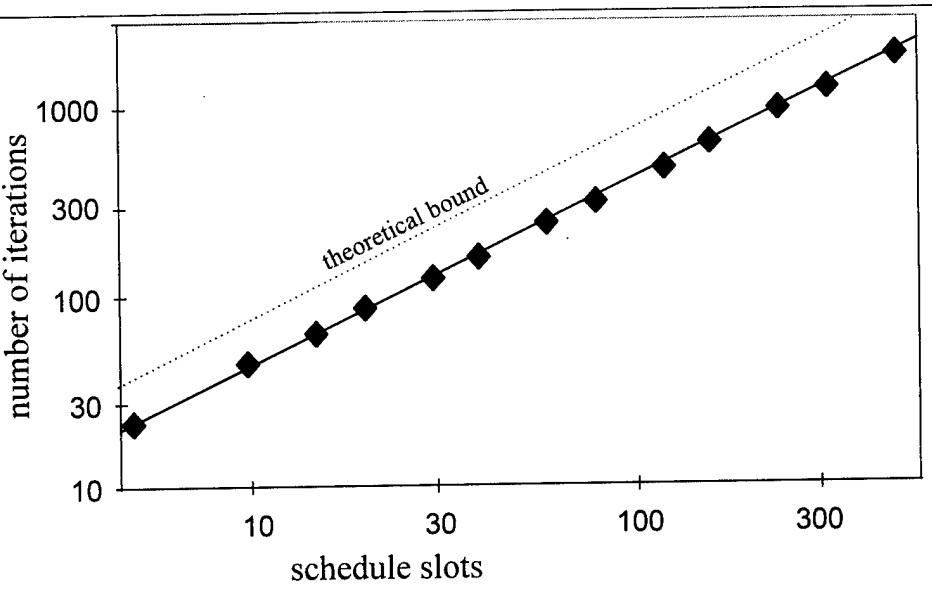


Figure 13. Total number of iterations required *vs.* the number of schedule slots for a set of fixed-lattice problems with 5 customers. The scale is logarithmic, and the dependence of the number of iterations on K^1 is evident.

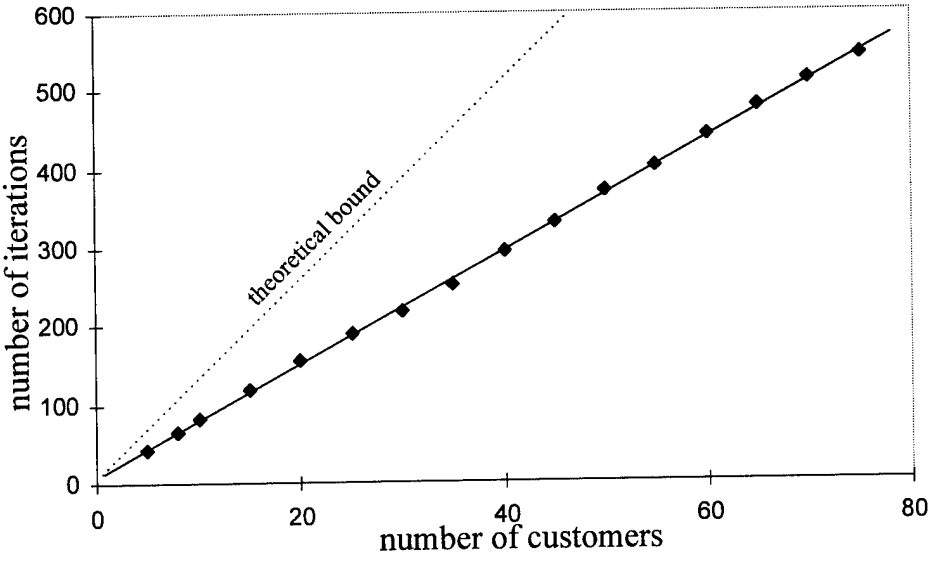


Figure 14. Total number of iterations required *vs.* the number of customers for a set of fixed-lattice problems with 8 slots. The dependence of the number of iterations on N^1 is evident.

The linearity of the number of iterations required by the fixed-lattice algorithm with respect to N and K is only matched by linearity with respect to program execution time if the time required for each schedule evaluation is independent of N and K. In fact, this is not the case for the proposed evaluation approach for non-identical customers. The number of matrix multiplications required for each evaluation is $O(N)$. The number of flops required for each matrix multiplication is dependent on the cube of the size of Q, which is $(\sum_{j=1}^{N} r(j))$, where $r(j)$ is the number of exponential service phases for the j^{th} customer. For iid services, the number of flops for each matrix multiplications is simply $O(N^3)$. Thus the run time of an optimization for this simplified case is $O(N^4)$. An example of this dependence is shown in Figure 15. Here, the same problem was used as that used to show the dependence of the number of iterations on N. The slope of the lower section of this log-log graph is 0.94, while the upper portion has slope of 3.87. This is highly suggestive of the conjectured N^4 dependence for higher values of N. For lower values, it is conjectured that the time consumed in preliminary calculations such as matrix exponentiation dominates the time spent in repetitive matrix multiplication.

Figure 16 shows the dependence of run time on K for the same set of problems that were used to show dependence of the number of iterations on K. The calculation of $(\exp(Q\Delta))^{(\tau_j - \tau_{j-1})/\Delta}$ is performed for each $j \in [2, N]$ by repetitive multiplication of $\exp(Q\Delta)$, so each schedule evaluation requires $\sum_{j=2}^{N}(\tau_j - \tau_{j-1})/\Delta = K - 1$ of these matrix multiplications, making the number of flops required for evaluation $O(K)$ (assuming the last schedule slot is occupied by the N^{th} customer). Therefore, the optimization run time is $O(K^2)$. This dependence is seen clearly for larger values of K.

This dependence of run time on N^4 and K^2 leads to run times of over an hour for $N = 100$ and $K = 10$ for two-phase services, using a 133 MHz Pentium$_{TM}$ processor. For larger problems, it may be advisable to pursue an evaluation algorithm

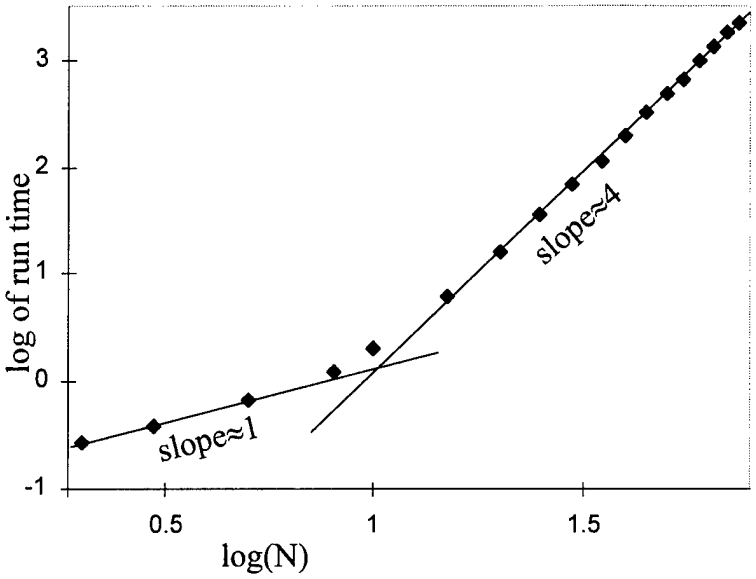


Figure 15. Dependence of run time on the number of customers. For larger N, the dependence of run time on N^4 is evident.

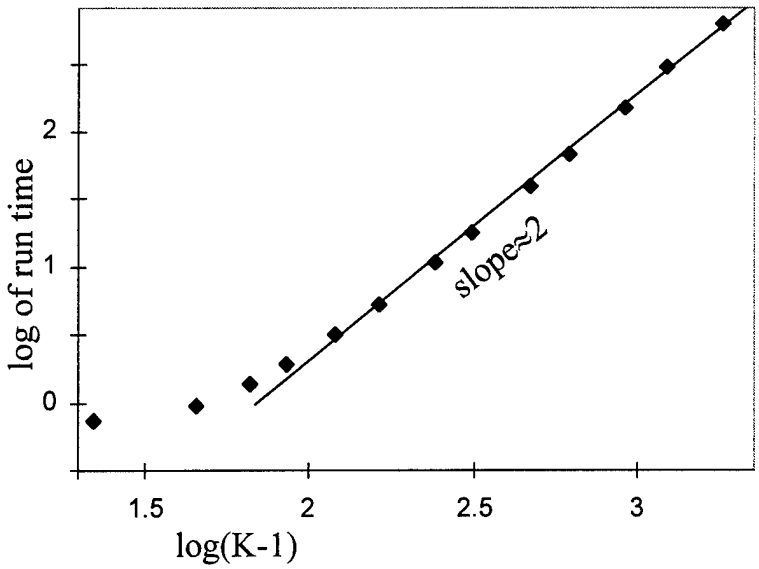


Figure 16. Dependence of run time on the number of schedule slots. For larger values of K, the dependence of run time on K^2 is evident.

that assumes iid or Erlang services. The fixed-lattice optimization algorithm will still prove effective for other cost evaluation approaches.

C.3 Comparison to Other Optimization Algorithms

It is worthwhile to compare the worst-case fixed-lattice search to a worst-case cyclic coordinate search for S_E. Compare Tables 14 and 16. The worst-case cyclic coordinate search must also evaluate $[\ 0\ \ j\ \ \ldots\ \ j\ ]$ for $j \in [0, K-1]$. Between these K "markers", there can be at most $N-2$ successes and one failure, and no failure is possible between the last two markers, making a total of $(K-1)(N-1)+K-2+1 = N(K-1)$ evaluations. In the limit as N approaches infinity, the ratio of the worst-case number of fixed-lattice evaluations to the number of cyclic coordinate evaluations is $2-\frac{1}{K-1}$, so the fixed-lattice search can be thought of as being approximately half as efficient as a simple cyclic coordinate search on a lattice. Of course, there is no assurance a cyclic coordinate search will converge to the optimum for these schedule problems, but this comparison establishes a rough "price tag" on the desire to obtain the precise lattice optimum, rather than an approximation.

If only an approximation to the optimum is desired, a reasonable approach is to employ a nonlinear program (NLP), obtaining an approximation to the continuous solution and then choosing the closest lattice schedule. For a number of problems, quasi-Newton and Nelder-Mead searches were performed, using the equispaced schedule as a starting point. Both searches were set to halt when the search was narrowed to a region less that Δ in width in each direction. IMSL$_{TM}$ routines UMINF and UMPOL were used. The results were compared to those of a fixed-lattice approximation algorithm to find S_E and S_L, starting at some schedule S' for which there was reasonable assurance that $S' \geq \hat{S}$ or $S' \leq \hat{S}$.

For a problem with 10 slots, 6 customers, identical exponential services with mean of Δ, and all cost coefficients set equal, the Newton-Raphson algorithm re-

Table 16. Example of a worst-case search for S_E using a cyclic coordinate algorithm. $N = 4$ and $K = 5$

[**0** **0** **0** **0**]	start
[0 0 0 1]	success
[0 0 0 2]	failure
[0 0 1 1]	success
[**0** **1** **1** **1**]	success
[0 1 1 2]	success
[0 1 1 3]	failure
[0 1 2 2]	success
[**0** **2** **2** **2**]	success
[0 2 2 3]	success
[0 2 2 4]	failure
[0 2 3 3]	success
[**0** **3** **3** **3**]	success
[0 3 3 4]	success
[0 3 4 4]	success
[**0** **4** **4** **4**]	success

quired 53 evaluations and the Nelder-Mead algorithm required 130 evaluations to find an approximate solution to the optimal schedule. The fixed-lattice approximation was started from [0 0 2 4 6 8] and required 19 evaluations. Even when the lattice approximation was started from the earliest possible schedule, it required only 53 evaluations, still competitive with NLP methods. NLP approaches tend to perform better compared to the fixed-lattice algorithm as the lattice size decreases, while the fixed-lattice approximation tends to perform relatively better when the number of customers is greater.

C.4 Comparison of the Fixed-Lattice Algorithm to Liao's Algorithm

Liao's scheme is the only other known approach to finding the lattice optimum of an appointment system [96, 95, 97]. He employed an effective branch-and-bound technique, using the solution of the associated dynamic scheduling problem as an upper bound at each stage of the static problem. This leads to efficient solutions for scheduling problems with iid Erlang-r services.

Liao's approach makes powerful use of recursive approaches for finding the cost of one schedule from another in which only one customer has been shifted one slot. This recursive approach to evaluation, which requires far less calculation than a full evaluation when services are iid Erlang-r, does not appear to be easily effected for the case of idd Coxian services with no-shows. This is an area for future research, since Liao's optimization scheme could be quite effective.

Because Liao's recursive approach was such an integral part of his conception, it is difficult to compare the two methods. Table 17 shows the effectiveness of his algorithm in terms of the number of partial schedule evaluations, while it shows the fixed-lattice algorithm effectiveness in terms of full evaluations. In light of the above discussion, however, this seems a good basis for comparison; it seems reasonable to suppose that either his recursive evaluation approach could be employed in the fixed-lattice optimization or that such a scheme would have to be abandoned in his algorithm if it were to encompass idd Coxian distributions with no-shows.

The problem used by Liao used iid exponential services with mean of 1.6, with no schedule horizon and essentially an overtime point of zero [97]. The ratio of the cost of overtime to waiting time was 3. It is unclear how he modified the lattice size between runs, so the fixed-lattice algorithm was run using a lattice size of $5/(K-1)$. Because this particular problem results in very fast evaluations for the fixed-lattice algorithm, the overtime point was shifted to 5.0 for a more representative comparison.

In the first two runs below, it can be seen that Liao reports requiring more evaluations required than the total number of feasible schedules. This discrepancy appears due to his not recognizing the first customer's arrival time is fixed, which increases the number of schedules his approach had to fathom by a factor of $N+K-1$.

Liao's program apparently was unable to handle $K > 24$ or $N > 0.2K$ for large values of K, due to excessive run times on an Intel 80386 processor, so comparisons are limited [97]. Liao's approach is apparently superior in the case of $N = 8$, $K = 6$, but in the other runs, the fixed-lattice algorithm appears much more efficient. This

is true despite abnormally large enumeration phases in the last two cases (112 and 50 evaluations, respectively). Regardless of comparative effectiveness for these small problems, it is clear that as N and K increase, the fixed-lattice algorithm becomes relatively more effective, and that even if Liao's approach were superior for some set of smaller problems, the fixed-lattice approach would surpass it at some problem size.

Table 17. Comparison of fixed-lattice and Liao's results [97]

N	K	fixed-lattice time (seconds)	fixed-lattice algorithm	Liao's algorithm	total possible schedules
2	12	1.98	5	15	12
3	15	2.69	21	56	30
3	18	3.30	25	75	171
4	6	2.36	25	31	56
4	21	5.06	53	191	1771
4	24	7.35	63	351	2600
6	8	3.29	47	65	792
8	6	3.51	62	34	792
8	10	6.48	106	188	11440
10	8	9.44	186	268	11440
12	10	11.04	176	622	167960

C.5 Effectiveness of the Sequencing Algorithm

Each iteration of the sequencing heuristic proposed in Chapter V is based on determining the optimal schedules for each of the schedules that can be obtained from the current optimum by a pairwise swap, and selecting the one with lowest cost. The number of schedule optimizations required to reach the conjectured optimum is therefore the product of the number of iterations required and $\binom{N}{2}$, the number of swaps per iteration. In the worst case, the number of iterations required could be factorial. However, as seen from Table 12, the average number of iterations is rather low, and the maximum number of iterations required is substantially less that the number apparently possible. The number of iterations appears exponentially distributed, and a large number of iterations is seldom encountered. This suggests

that, while the sequencing problem is almost certainly NP-hard, the sequencing algorithm is usually polynomial with respect to N. This is a similar result to that obtained for the scheduling problem. It is also analogous to the application of simplex methods to linear programming problems; the problem is NP-hard, but the algorithm usually performs in polynomial time [15].

that, while the sequencing problem is almost certainly NP-hard, the sequencing algorithm is usually polynomial with respect to N. This is a similar result to that obtained for the scheduling problem. It is also analogous to the application of simplex methods to linear programming problems; the problem is NP-hard, but the algorithm usually performs in polynomial time [15].

Appendix D. Sensitivity Analyses

This section empirically explores the sensitivity of the optimal schedule and cost to cost coefficients, no-show rate and service distribution moments. The examples shown are all excursions from a basic schedule with five customers constrained to arrive within a time period 5 units long, divided into 101 slots. The start of overtime is set to 5.0. Unit costs for expected waiting times and idle times are set to 1. As a baseline, customers have a show probability of 1, and service distributions are iid with mean of 1.0.

The plots of the examples, Figures 17 through 20, share several conventions. For the optimal schedule plots, the arrival time for the first customer is omitted, since it is always zero. The data for each customer in the optimal schedule plots are connected by straight lines. These lines are not data fits and serve merely to assist the reader in visually assembling the data. On the other hand, the data in the optimal cost and expected waiting time plots are shown as unconnected points. Any lines that appear to connect these data are either empirical or theoretical fits, as discussed in each section.

D.1 Dependence of Optimum on Cost Coefficients

The plots in Figures 17 and 18 exemplify the dependence of optimal cost and schedule on the individual cost coefficients. This dependence has already been discussed in passing in Chapter IV and Figure 7. In this example, the customer services are iid Erlang-4 distributions with mean of 1.0, and show probabilities are all 1.0. Cost coefficients other than the one in question are held constant at 1.0.

As the relative size of c_6, the overtime coefficient, decreases in Figure 17, the optimal schedule and cost approach the limit defined by setting $c_6 = 0$. As the relative size increases without bound, the optimal schedule is forced to the earliest possible schedule. The cost due to the expected waiting times in this case would

approach $1+2+3+4 = 10$, while the cost due to overtime would approach $\mathrm{E}[W_6]c_6 = 0.444c_6$. A least-squares fit of the upper four cost data points yields $C = 4.5+0.444c_6$. The discrepancy in intercept is due to the fact that the optimal arrival times have not yet reached zero.

The three plots in Figure 18 exemplify the dependence of optimal schedule and cost on the cost coefficient of a single customer. Each plot can be thought of as being divided into three regions: $c_3 < 1$, $1 < c_3 < 10^4$, and $c_3 > 10^4$. In the first region, the plots show that as the importance of the third customer decreases, the schedule and cost approach that of a four-customer system with all cost coefficients equal, but with the second customer having twice as many phases, and thus twice the expected service. This region is characterized by all optimal arrival times and expected waiting times but the third being relatively insensitive to c_3. Apparently in this region, changes in the position of the third customer have little effect on the optimal expected waiting times, and thereby positions, of subsequent customers.

In the region $1 < c_3 < 10^4$, the optimal arrival times of the fourth and fifth customers shift and begin to become constrained by the horizon. This causes an appreciable increase in each expected waiting time but the second, and the cost increases. A least-squares fit to a power function yields $C = 0.18c_3^{0.17}$ with good precision. An analytical reason for this fit is not forthcoming.

Only in the region $c_3 > 10^4$, when the optimal arrival times of the other customers are completely constrained by the horizon, do the optimal τ_2 and $\mathrm{E}[W_2]$ shift appreciably. The fact that the arrival times of the fourth and fifth customers are fixed at the horizon suggests that $\mathrm{E}[W_4 + W_5 + W_6]$ approaches some amount in excess of $1 + 2 + 3 = 6$, which can be seen in the second and third plots. Since the expected wait of customer 3 is $2.8 \cdot 10^{-3}$ at this extreme position, one would expect the cost to be approximately $C = 2.8 \cdot 10^{-3}c_3 + 6 + 3 \cdot 2.8 \cdot 10^{-3} = 2.8 \cdot 10^{-3}c_3 + 6.0084$. Regression yields $C = 2.8 \cdot 10^{-3}c_3 + 6.33$. The discrepancy in this case is due to the second customer's expected wait increasing as its optimal arrival time decreases.

When the second customer's optimal arrival time reaches zero, the expected form is $C = 2.8 \cdot 10^{-3} c_3 + 7.0$, which was indeed observed to good accuracy.

D.2 Dependence of Optimum on Show Probability

The plots in Figures 19 and 20 exemplify the dependence of optimal cost and schedule on the show probability. The customer services are iid Erlang-4 distributions with mean of 1.0. When show probabilities for all customers are reduced simultaneously, the optimal arrival times decrease regularly. For this example, as γ is decreased from 1.0 to 0.6 (the typical range for medical applications), the decrease is between 15% and 17% for each customer's optimal arrival time. For a service mean of 4.0, a choice for which server overtime plays a larger role, this shift increases to about 25%.

The optimal cost under this shared show probability is seen to fit very closely the curve $C = 0.99\gamma^2$ (correlation coefficient of 0.9996). The form of the equation for this particular example is misleadingly simple. In general, a good approximation can be obtained by the 2-parameter form $C = a_1\gamma + a_2\gamma^2$, with correlation coefficient over 0.999.

Further, while it is tempting to seek a simplistic explanation for the above equation fit, the reader should be warned that linear regressions over a number of examples indicate that individual expected waiting times of customers are better fit by $\mathrm{E}[W_j] = a_2\gamma^2 + a_3\gamma^3$ than by the form above.

Figure 20 shows the effect on the optimal schedule and cost of varying γ_3, the show probability of the third customer, holding other show probabilities constant at 1.0. As γ_3 decreases, the arrival time of customer 3 also decreases slowly, as might be expected. The other customers shift gradually to their optimal arrival times in a four-customer schedule, as does the cost.

The third customer's position as $\gamma_3 \to 0$ can be understood by realizing that its scheduled arrival time has negligible effect on the scheduled arrival times of the other

customers, but when it does arrive, the costs due to both $\mathrm{E}[W_3]$ and $\mathrm{E}[W_4]$ increase (as do those of subsequent customers, but to a negligible extent). The optimal τ_3 will balance these costs. Compare this case to the first plot in Figure 18, where as $c_3 \to 0$, the contribution of $\mathrm{E}[W_4]$ is nearly constant, while that of $\mathrm{E}[W_3]$ goes to zero, forcing the optimal τ_3 to that of τ_2. Note that the near-constant contribution of $\mathrm{E}[W_4]$ as $c_3 \to 0$ forces the optimal τ_4 and τ_5 much later than the optimal τ_4 and τ_5 when $\gamma_3 \to 0$.

The customers adjacent to the modified customer will be most affected by shifts in its show probability; as the show probability changes from 1.0 to 0.6, the fourth customer's optimal arrival time is reduced by 8.5%, while the second customer's arrival time is increased by 4.8%. Changing each customer's mean to 2.0 or 0.5 reduces the magnitude of these shifts, so in some sense, this is a worst case. The solid curve on the cost plot is a quadratic fit (correlation coefficient of 0.9998).

D.3 Dependence of Optimum on Service Distribution Mean

Figure 21 exemplifies the effect as the means of each service distribution are varied together. As the mean tends to zero, the optimal schedule tends toward equi-spacing, and the cost tends to zero. As the mean increases, each customer's optimal scheduled arrival time tends toward the schedule horizon; there is little possible advantage accrued by scheduling them earlier, since it is almost certain that the first customer's service will be greater than the horizon. The optimal cost in this case tends toward the worst-case cost when a horizon is imposed, which in the case of $c_1 = c_2 = \cdots = c_{N+1} = 1.0$ is the product of the service mean and $N(N+1)/2$. This asymptote is shown as a solid line in the log-log plot of cost.

Figure 22 exemplifies the effect as the mean of only one customer (in this case the third) is varied. As the mean decreases, the optimal schedule and optimal cost tend toward that of a four-customer system. The horizontal solid line in the log-log cost plot represents this four-customer cost. As the mean increases, the fourth

and fifth customers are forced to the horizon, while the second and third find an equilibrium. Beyond some point, the cost is dominated by the contribution of the third customer to the expected waiting times of the subsequent customers and the server overtime, each of which tends to the mean of the third customer reduced by the difference in optimal arrival time between the subsequent customers (5.0) and the third customer (1.5). The solid curve in the log-log cost plot, $C = \text{mean} - 3.5$, represents this asymptotic cost.

D.4 Dependence of Optimum on Standard Deviation of Service Distribution

Figures 23 and 24 exemplify the dependence of optimal cost and schedule on the standard deviation of the service. The mean for each customer was held at 1.0. Each service distribution was modeled as an Erlang-r distribution if c, the coefficient of variation, was less than 1.0, and as a Coxian-2 distribution otherwise. In the latter case, the third moment was set at its minimum value.

The behavior of the optimal schedule as each customer's standard deviation is varied is complex; starting from deterministic services, the arrival times are first shifted later when σ is increased, but as later arrivals are constrained by the horizon, the trend for earlier customers reverses. As σ increases, it dominates other considerations, and customers are polarized, arriving either at the earliest possible time or the latest.

When the schedule is close to deterministic, the cost is linear with respect to variance (slope=3.65, intercept=0.02, and correlation coefficient of 0.999 for $c < 2$ in this example). As the schedule becomes polarized, with each customer arriving at either the latest or earliest possible time, the cost approaches the maximum schedule cost, as discussed above. Here, the mean is constant at 1.0, and that maximum cost is 15.0. The solid curve in the cost plot is the least-squares fit of the upper five data points to $C = a_1 - a_2\sigma^{a_3}$, which yields $a_3 = 15.004$.

As only the standard deviation of the third customer (σ_3) is modified, holding others constant at 2.0, Figure 24 shows the dependence. The optimal schedule is relatively insensitive to σ_3 at smaller values but eventually the third and subsequent customers tend to some limiting optimal schedule. That schedule has the third and fourth customers arrive simultaneously; the advantage accrued when the service of the third customer is small outweighs the disadvantage when it is large, so an increase in variability drives the customers closer. The cost at small σ is again linear in σ (slope of 1.46 and correlation coefficient of 0.999). As σ increases sufficiently, the cost approaches that of the limiting 4-customer schedule obtained by removing customer 3 (0.78), plus 3.0. If it were certain that the third customer's service would be sufficient to cause each subsequent customer to wait, insertion of the third customer just before the fourth would add six units of cost. The probability of this occurrence is close to 50%, which results in the 3.0 added units of cost.

D.5 Dependence of Optimum on Service Distribution Skewness

When $c = 1.0$, the optimal schedule and cost are quite insensitive to the third moment. Define γ as the skewness of the distribution for this section only. As γ is varied over its entire possible range (*cf.* Section F.2) while holding mean and variance constant at 1.0 and 2.0, only a 6.5% change in cost is observed. The optimal scheduled arrival time for each customer varies at most 25% of the service mean over this range.

Figures 25 and 26 show the dependence of optimal cost and schedule on service distribution skewness when c has been increased to 1.73. These plots exemplify the dependence on γ for higher coefficients of variance. Moments were matched using a Cox-plus-Erlang-r distribution, as described in Section F.9, and r was arbitrarily limited to 16. For this reason, the smallest γ obtainable was 0.22, which is the minimum value displayed on the plots.

As all customers' skewnesses are increased from zero together, the optimal schedule approaches some limit. Above $\gamma = 0.8$, the schedule is quite insensitive to γ.

As the skewness increases, the cost decreases. This can be understood by noting that, in order to increase the skewness of a given distribution while maintaining the same mean and variance, it is necessary not only to shift a portion of the probability mass farther to the tail. It is also necessary to shift a larger portion of the probability mass lower. This second action dominates the first in the outcomes illustrated.

The cost approaches 3.77 as the skewness is decreased to zero and 7.16 as the skewness increases without bound. These values were determined by a least-squares approach to fitting the lower data to $C = a_1 + a_2\gamma^{a_3}$ and the upper data to $C = a_1 + a_2 \exp(a_3)$, and the solid curves in the cost plot show these fits. An analytical explanation of these limits is not apparent.

Figure 26 shows the dependence of the optimum on a single service skewness. When the third skewness is near zero, there is a possibility that the service will be very small. As a result, the fourth customer's optimal arrival time converges to that of the third. Again, for skewnesses over 0.8, the schedule is quite insensitive to changes in skewness.

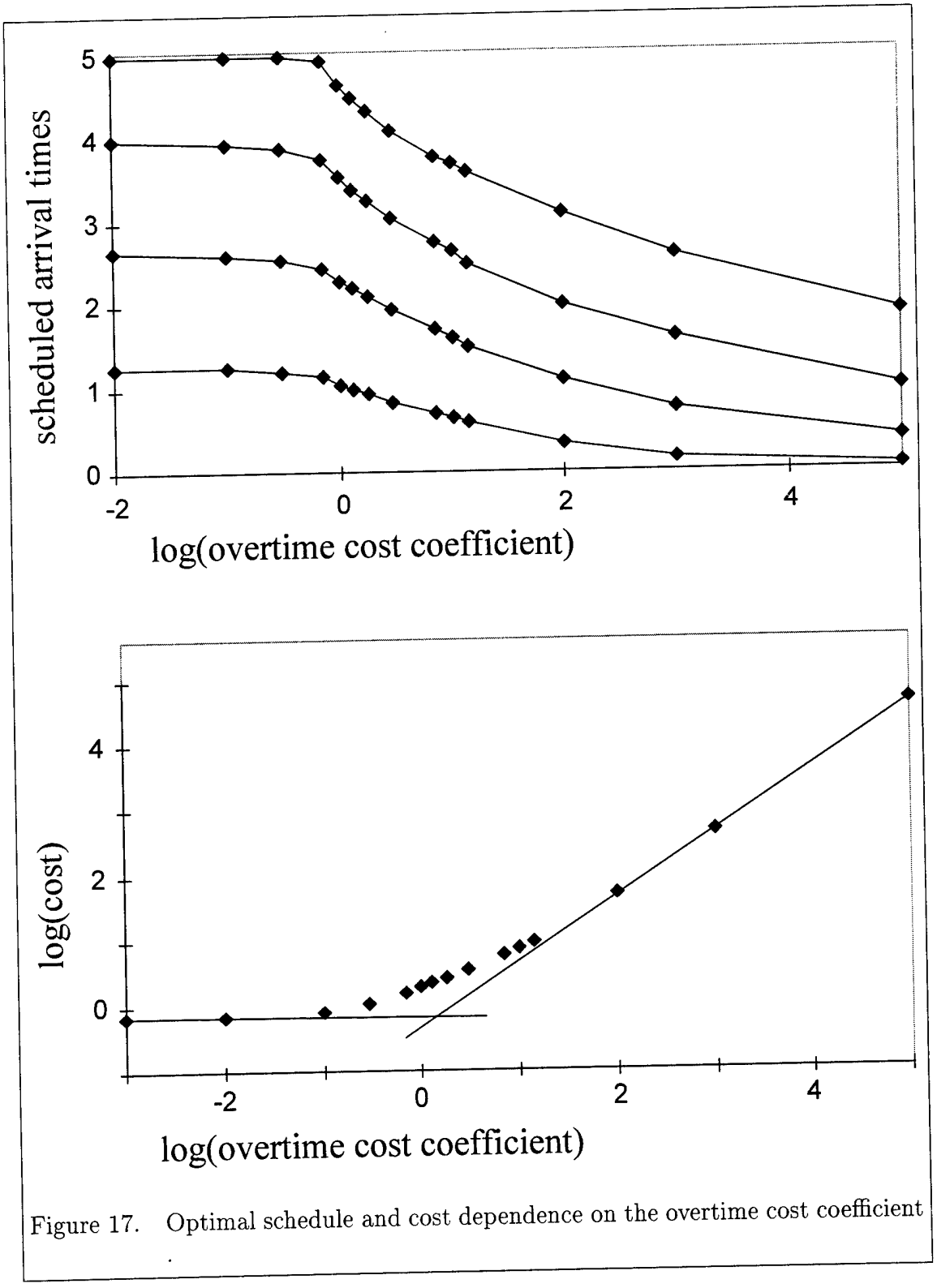


Figure 17. Optimal schedule and cost dependence on the overtime cost coefficient

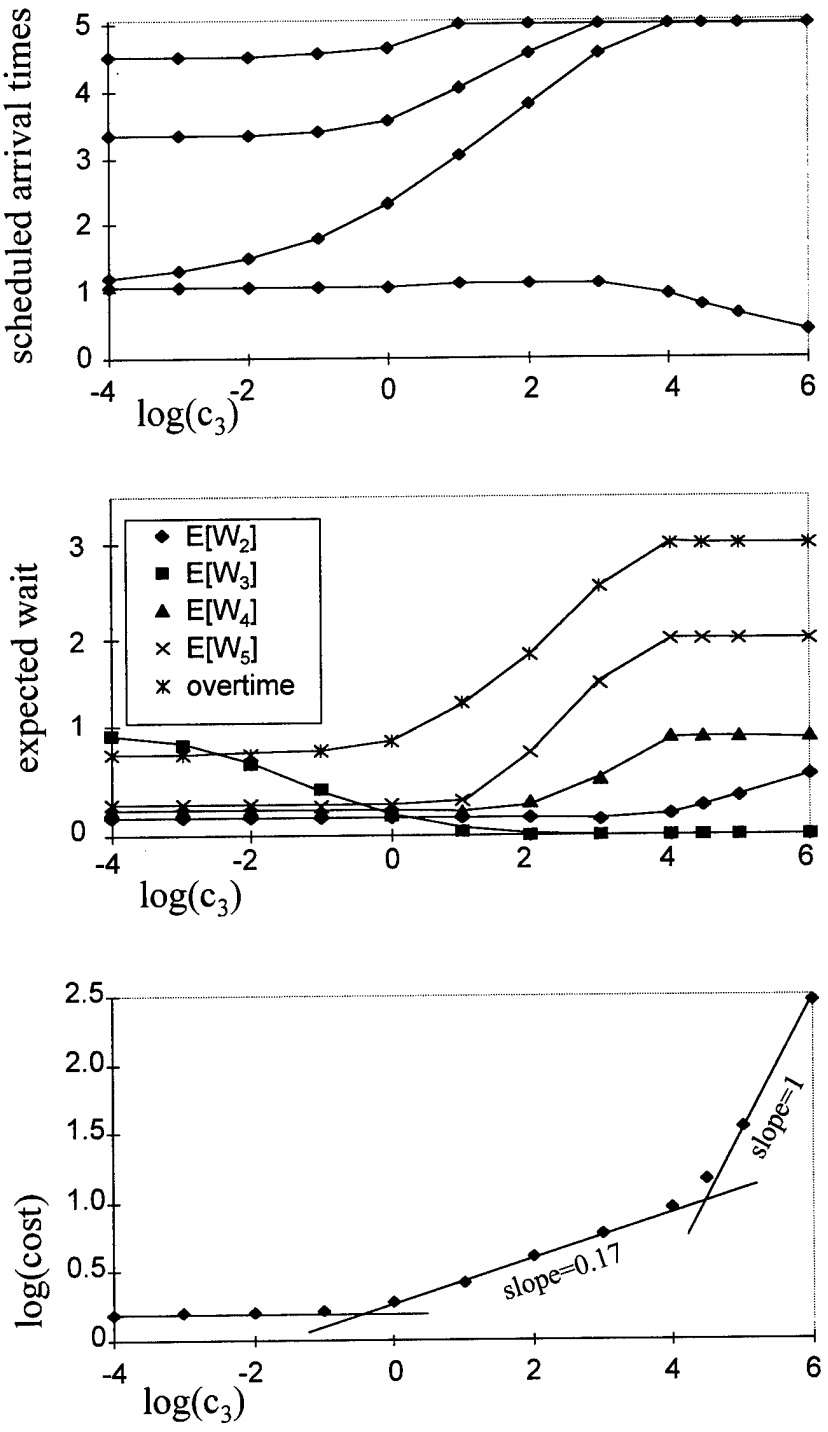


Figure 18. Optimal schedule and cost dependence on the cost coefficient of a single customer. Here, c_3 is varied while the other cost coefficients are held constant.

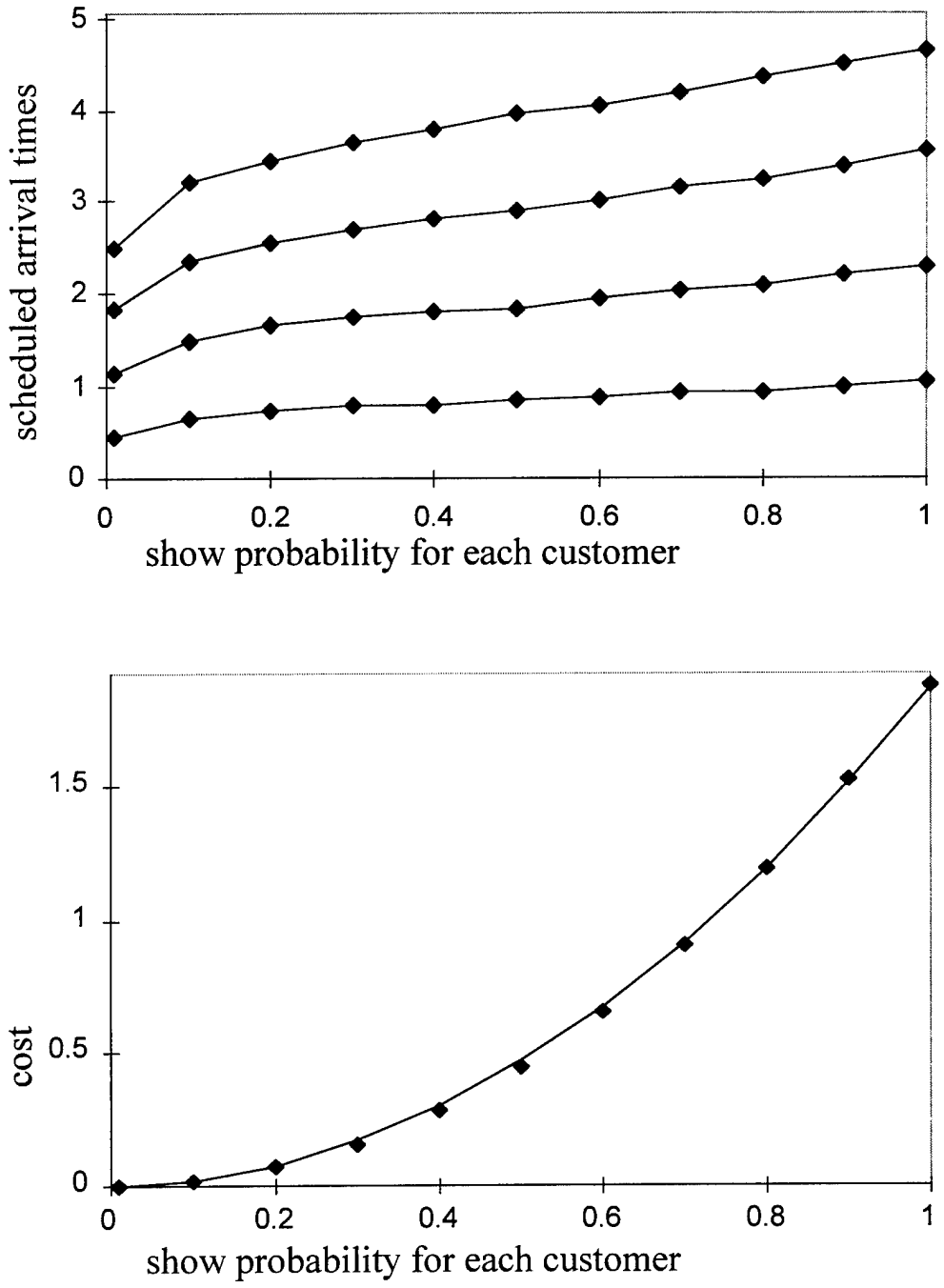
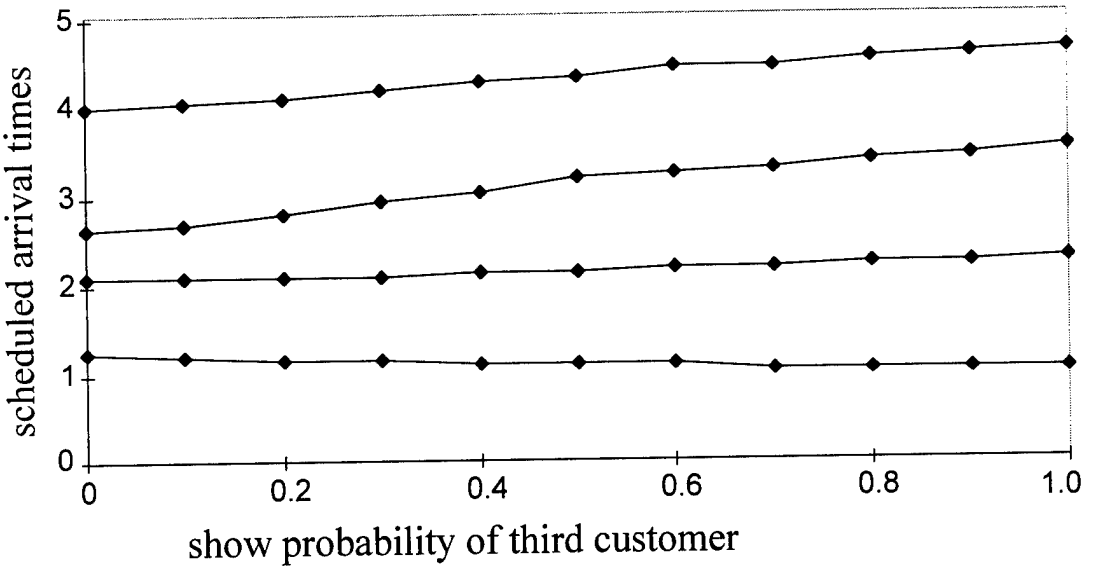


Figure 19. Optimal schedule and cost dependence on show probability for all customers. The same show probability is shared by each customer.



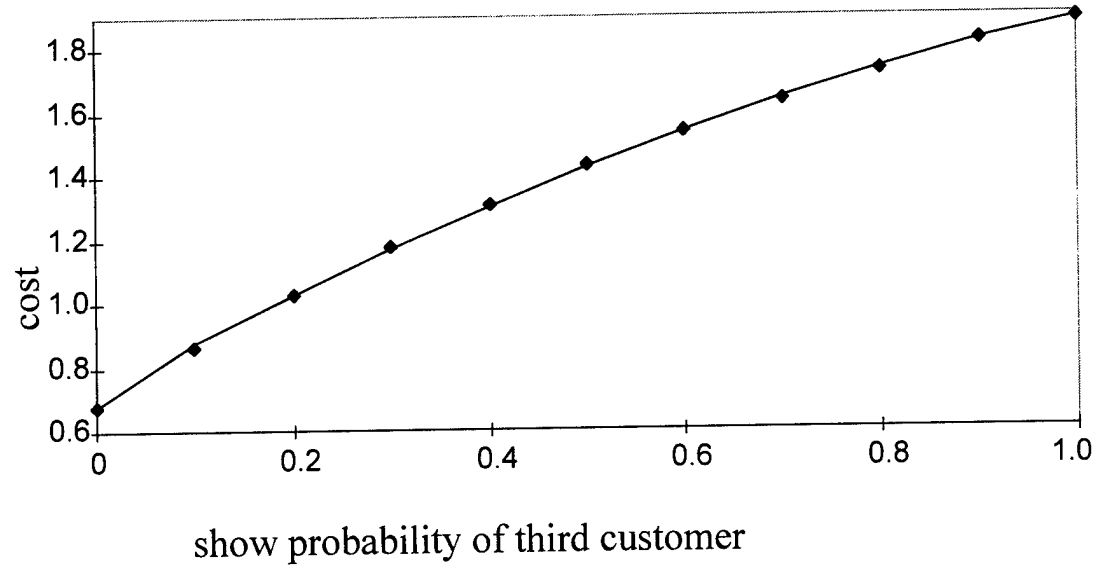
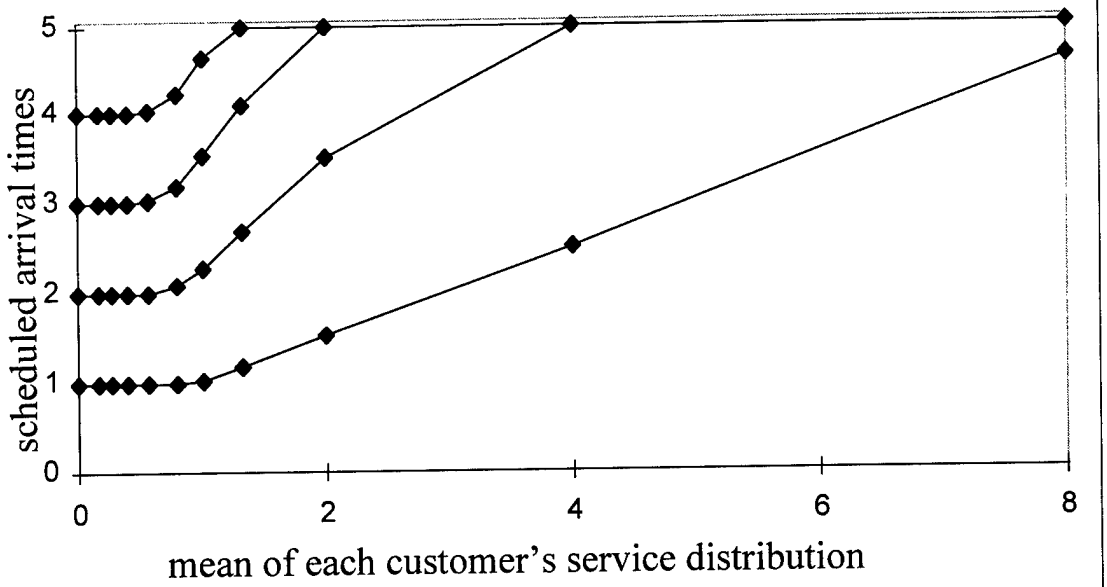


Figure 20. Optimal schedule and cost dependence on show probability of a single customer. Here, γ_3 is varied, with all other show probabilities fixed at unity.



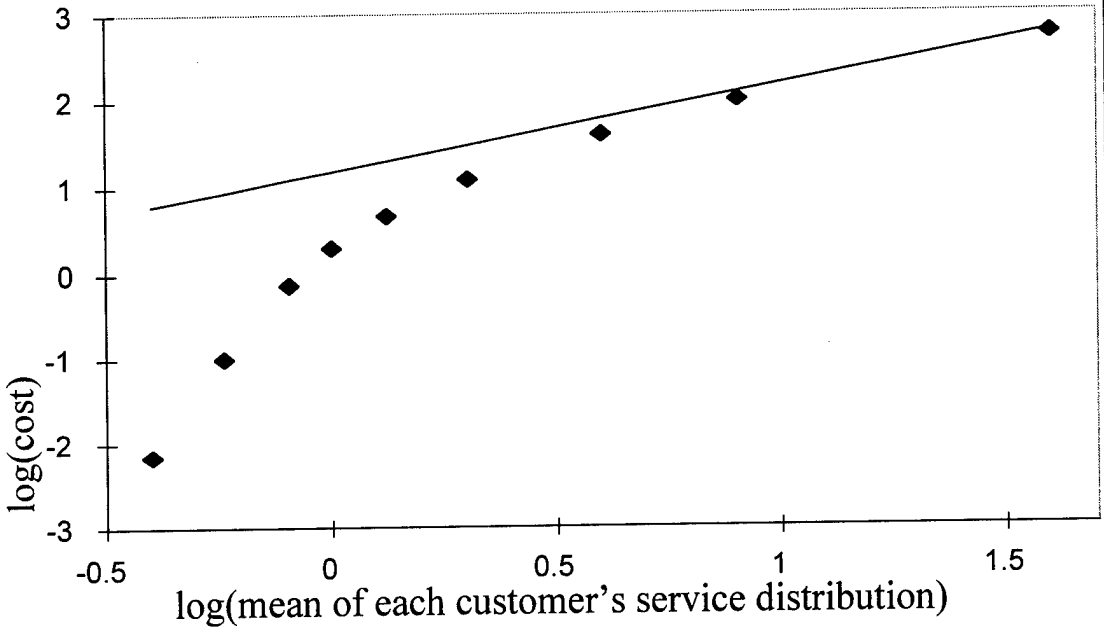


Figure 21. Optimal schedule and cost dependence on the mean of all customers. The same mean is shared by each customer.

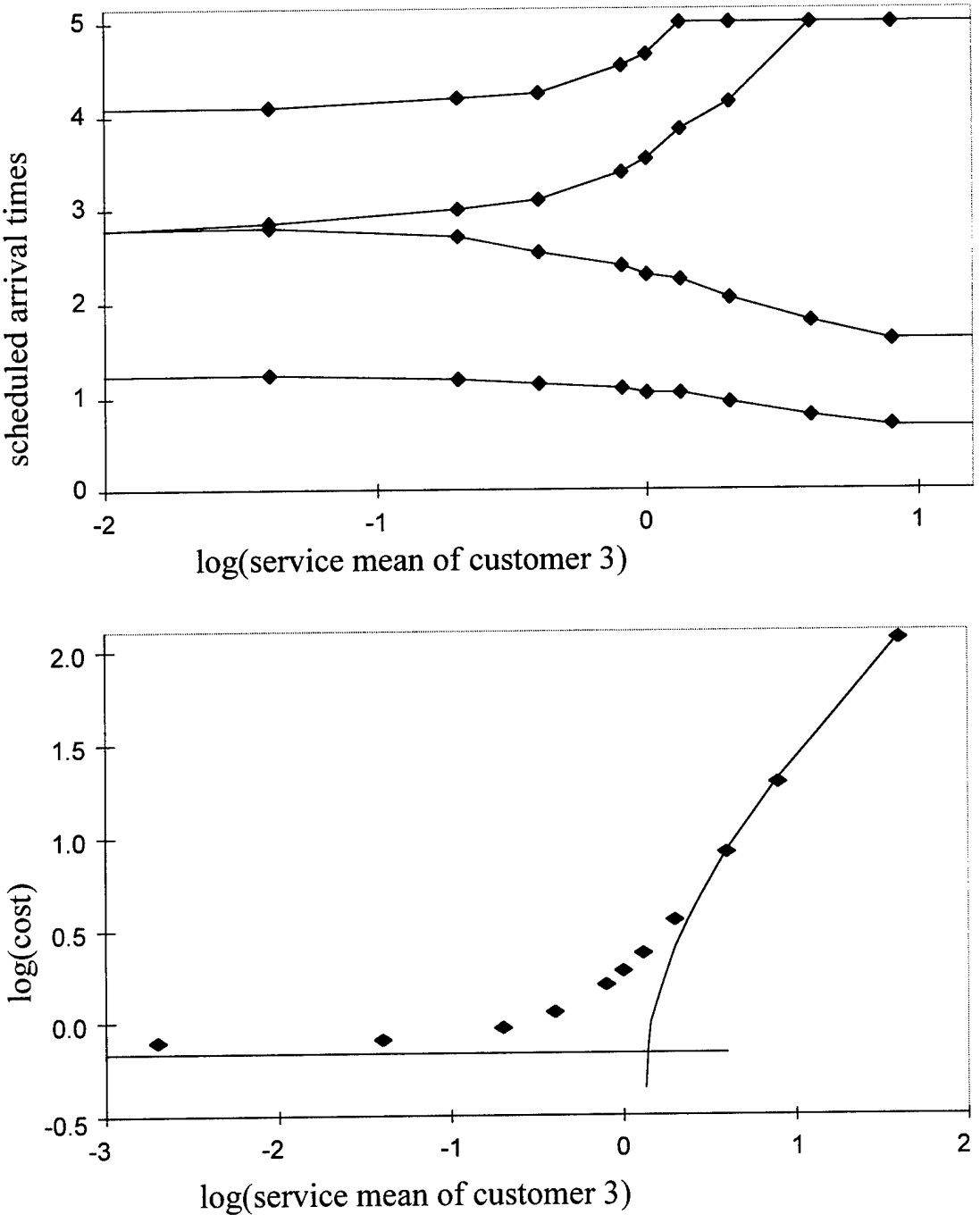


Figure 22. Optimal schedule and cost dependence on the mean of a single customer. Here, the mean of the third customer is varied while the means of the others are held constant.

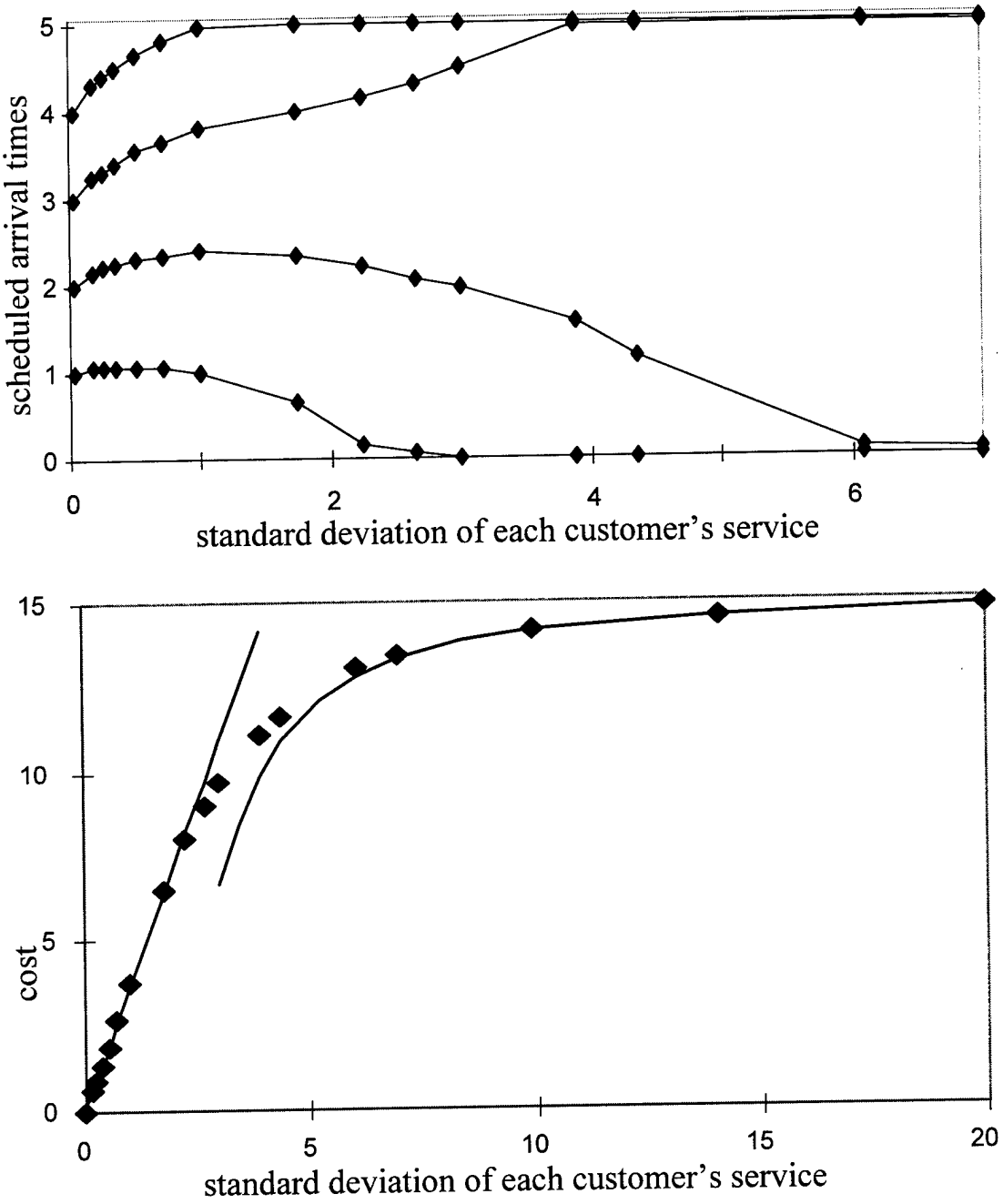
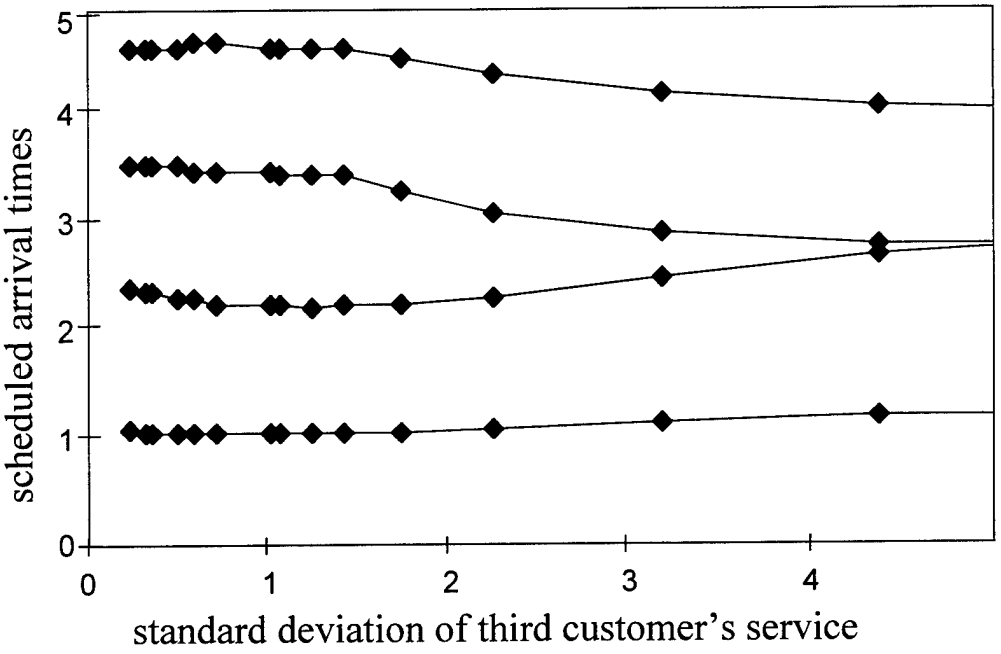


Figure 23. Optimal schedule and cost dependence on the service standard deviation of all customers. The same standard deviation is shared by each customer.



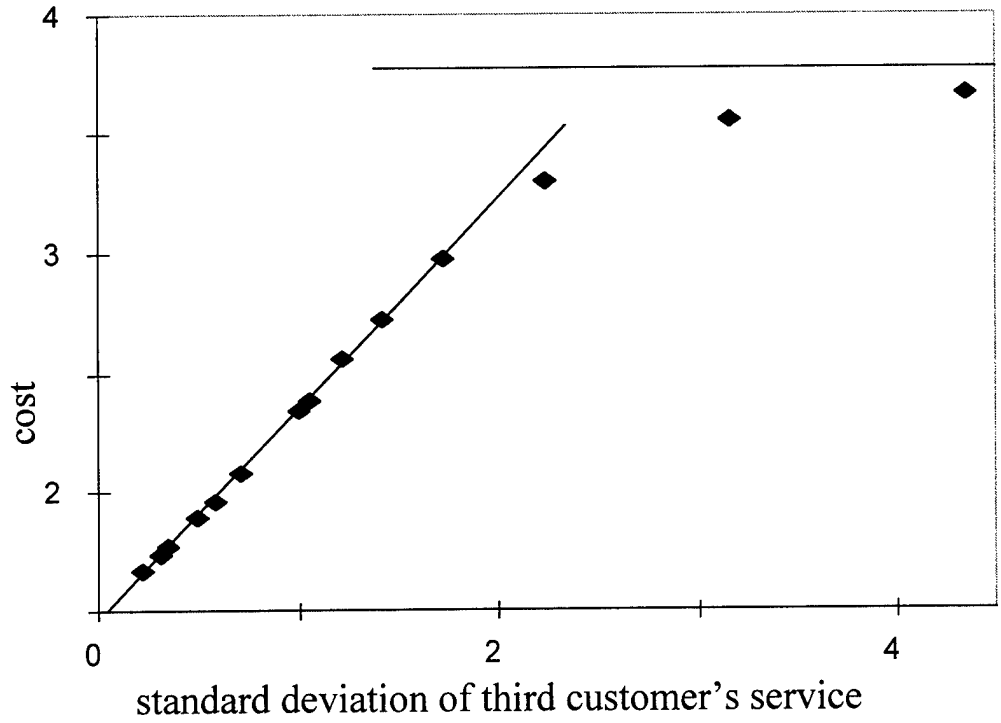
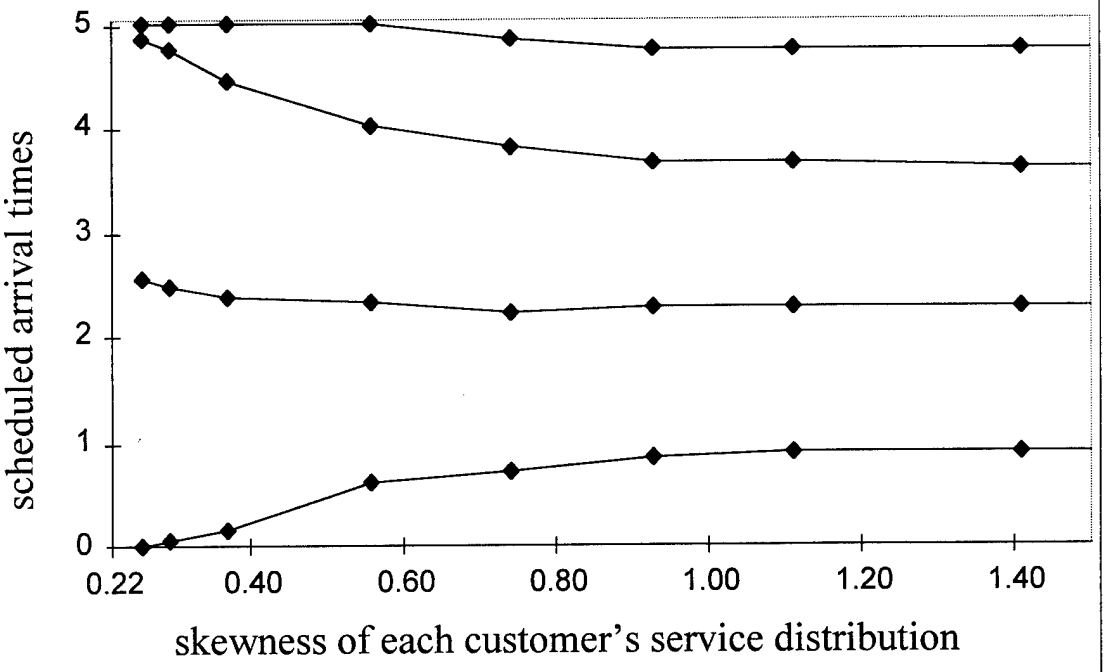


Figure 24. Optimal schedule and cost dependence on the service standard deviation of a single customer. Here, the standard deviation of the third customer's service is varied while the others are held constant.



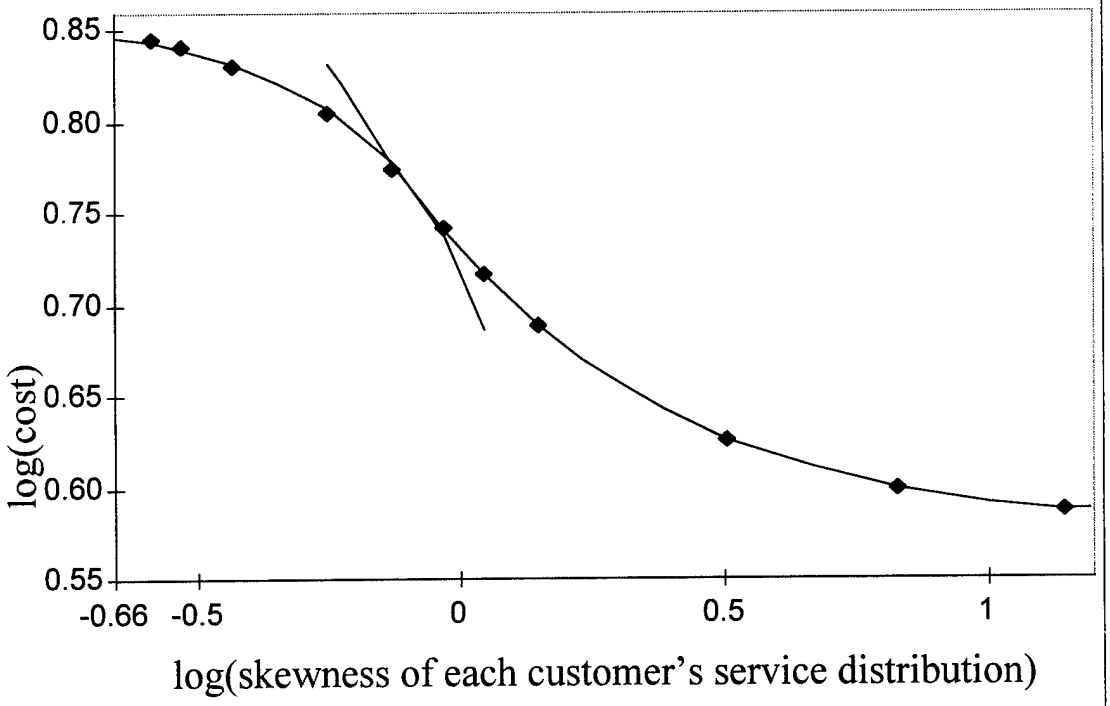


Figure 25. Optimal schedule and cost dependence on the service skewness of all customers. The same skewness is shared by each customer, and $c = 1.73$.

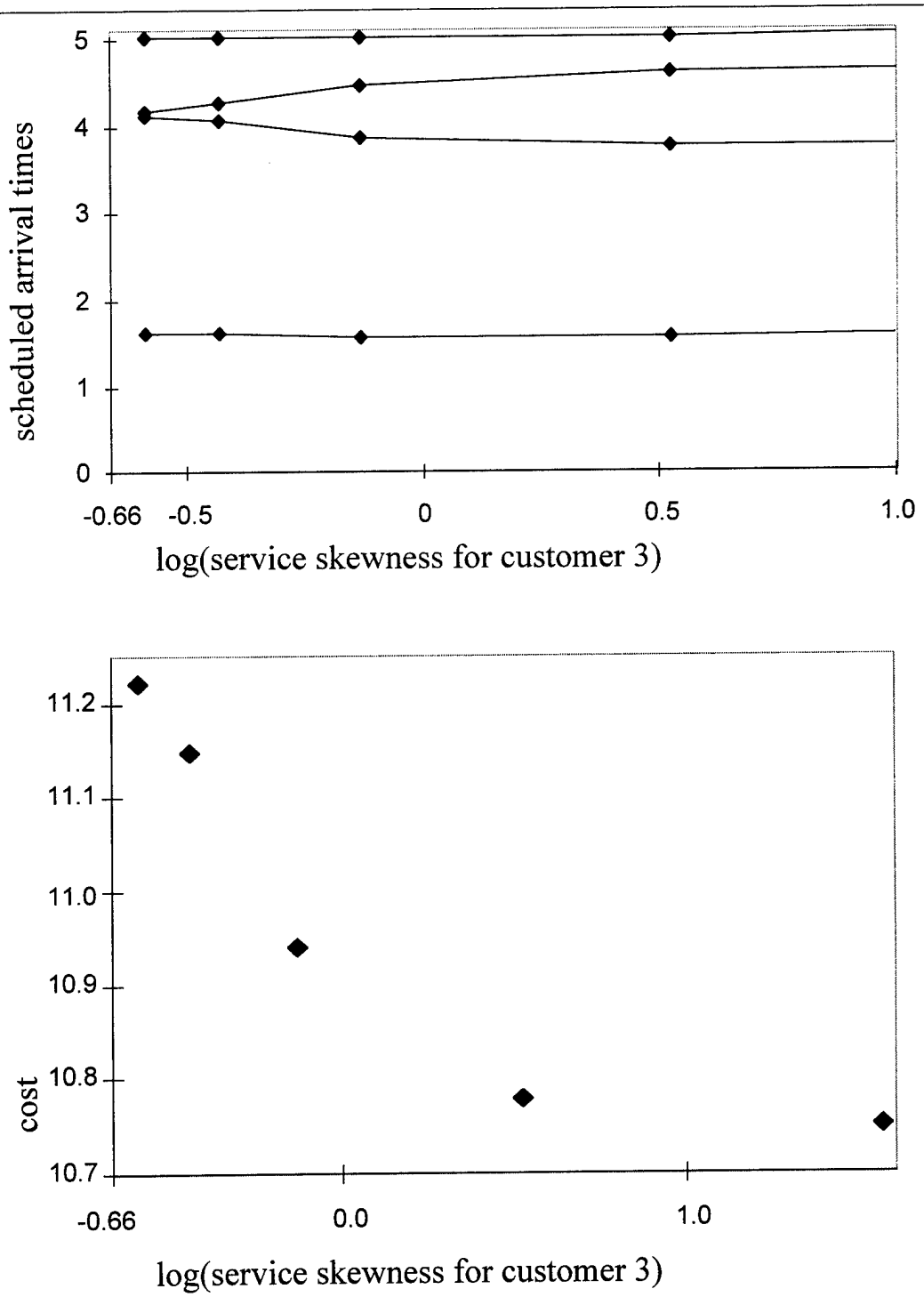


Figure 26. Optimal schedule and cost dependence on the service skewness of a single customer. Here, the skewness of the third customer's service is varied while the others are held constant, and $c = 1.73$.

Appendix E. Medical Scheduling Example

This appendix describes a study performed in 1996-97 by the author in conjunction with the Primary Care Clinic at Wright-Patterson AFB, OH. Appointment time data were collected over several months for 146 patients of a particular doctor.[1] From these data, appointment frequency, show rates, and service distributions were estimated for various classes of patients. Estimates of current cost and potential improvement from sequence and schedule optimization were then determined. The purpose was a preliminary determination of whether sequence and schedule optimization were appropriate and of value for this clinic.

E.1 Data

On a typical day, the doctor in this study would see 5 to 7 patients in the morning or afternoon over a period of approximately 170 minutes (horizon). Appointments were made by a clerk with no medical training, with slots being either 20 or 30 minutes in length, depending on whether the patient had been seen before. Classification of patients at the time of the appointment was based solely on whether the patient was a military dependent, retired military, or military on active duty, and all patients in this study were retired military or aged dependents. The average patient waiting time was about 7 minutes.

The doctor recorded the following data for 146 patients over 22 days: Appointment time, starting and ending times of the doctor's service, whether the patient showed, and the patient's classification. Data collection was stopped when the doctor was given orders to a new base. Although he continued to see patients for several months after this time, patient service time after this point was increased as the

[1]Dr Charles Beleny, Lt Col, USAF, Chief of Wright-Patterson Air Force Base Primary Care Clinic, provided access to clinic personnel and gave the author invaluable scheduling advice. Dr Robert Nardino, Maj, USAF, a physician in the primary care clinic at the time, collected the data using his own patients.

doctor sought to arrange follow-on care upon his departure. It was judged that the later data were not usable.

The doctor was asked to develop a patient classification scheme that could predict service time more accurately than the aggregate mean. Such a scheme was to be easy for the scheduler to use to classify patients over the phone. The scheme he settled on was to record the number of new complaints and the number of chronic complaints (complaints previously treated by this doctor) the patient requested medical care for at the time he/she made the appointment. The paucity of the data in this preliminary study necessitated the aggregation of the eight categories originally envisioned into the following three:

- Class A: fewer than three chronic complaints, no new ones (33 data points).
- Class B: more than three chronic complaints, no new ones (50 data points).
- Class C: at least one new complaint (56 data points).

One might classify this strategy of dividing the customers into classes as a variance reduction technique. By recognizing different classes, perhaps one could take advantage of additional information to establish more certainty in the service time of a customer. The second point is certainly valid, but this example suggests that the information gained is not contained in the variance, as some researchers have asserted or conjectured [31, 164]. The variance of the full sample is 77.1, while the variances for classes A, B, and C customers are 16.0, 106.7, and 66.7. Variance is not reduced enough to warrant major improvements in prediction. One should not think of this approach as a method of establishing customer service times more certainly, but rather as a method of taking into account the service time distribution information in some more complex way.

Histograms of the observed service times for these classes are shown in Figure 27. These histograms do not include no-shows.

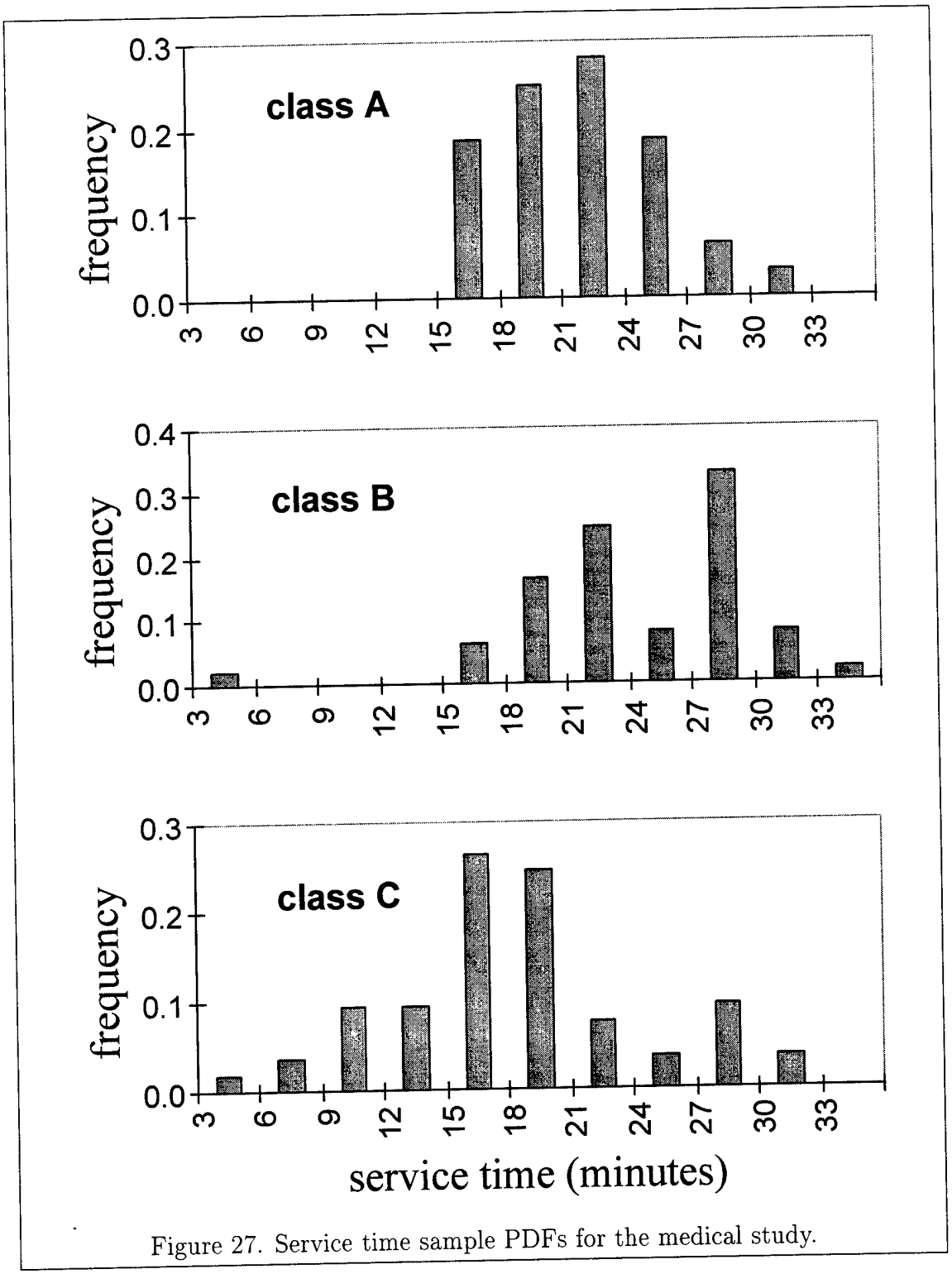


Figure 27. Service time sample PDFs for the medical study.

E.2 Assumptions

A number of assumptions were made as the study progressed. Some are listed here:

- The doctor's scheduling horizon is exactly 170 minutes.
- Six patients require scheduling each day.
- Patients cannot be scheduled more accurately than 10 minutes. Thus, 18 schedule slots are adequate.
- The unit value of each patient's time is the same, and the doctor's time is three times more valuable than that of the patients.
- The doctor is able to utilize idle time during the schedule horizon productively, so the value assigned to his time during the scheduling period is constant. Only his time spent after the end of the scheduling period is assigned a cost ($\tau_v = \tau_h$).

The doctor agreed these were reasonable assumptions, although it appeared difficult to ascertain a good value for the relative unit costs of patient and doctor time.

E.3 Analysis and Results

Sample statistics for the three service classes are shown in Table 18.

Table 18. Sample statistics for the medical study

class	mean	c^2	skewness	show rate
A	18.75	0.046	0.46	0.89
B	21.94	0.139	2.51	0.95
C	25.56	0.141	0.28	0.92

The best fits to the service distributions were found via a commercial software package to be truncated Erlang and truncated gamma distributions. However, because the coefficients of variation are so low, it was decided that matching the first two moments was sufficiently accurate. The Cox-plus-Erlang-r distribution described in Appendix F was fit to the data. Because the coefficient of variation is less than

1, this distribution is equivalent to a generalized Erlang distribution. The Coxian parameters in Table 19 were determined.

Table 19. Service distribution approximations for the medical example

class	Erlang phases	Coxian phase rate	Erlang phase rates	Cox transition probability
A	22	1.173	1.173	0.9994
B	8	0.360	0.360	0.9842
C	8	0.308	0.308	0.9820

Using these service distribution approximations, the optimization programs in Appendix H were used to obtain the optimal sequence and schedule for each combination of patients that might be scheduled on a given day, under the current policy. There are 28 possible combinations with 3 classes and 6 customers. Exhaustive enumeration of the $3^6 = 729$ possible permutations of these combinations was performed, and the optimization of each permutation took between 6 and 10 minutes on a 133MHz Pentium processor. Table 20 shows the results.

The greedy sequencing algorithm proposed in Chapter V was also employed to find the optimum for each combination above. For all but two combinations, the global optimum was obtained, regardless of the starting sequence used for the greedy algorithm. In the case of sequence AAABCC, of the 60 possible starting sequences, 16 yielded a suboptimal sequence with a cost that exceeded the optimum by 0.02 (0.3%), and 14 others yielded a suboptimal sequence that exceeded the optimum by 0.006 (0.09%). In the case of sequence AAAABC, 7 of the 15 possible starting sequences yielded a suboptimal sequence that exceeded the optimum by 0.11 (3.1%). However, although these errors may be significant, neither of the combinations turned out to play a role in the solution to be proposed.

The relative frequencies of classes A, B, and C in the data sample are 0.237, 0.360, and 0.403, respectively. If these frequencies hold in the long run, a reasonable appointment strategy would be to employ only some subset of the optimal sequences and ensure that subset allows the scheduler to cope with small variations from the

Table 20. Optima for each combination for the medical scheduling problem. The possible combinations for this problem, their frequency of occurrence under random selection, and the optimum sequence and schedule, given that combination.

	random	optimum		
combination	frequency	permutation	schedule	cost
ABBCCC	0.1206	ACCCBB	0 20 50 80 110 140	16.94
ABBBCC	0.1077	ACBCBB	0 20 50 80 110 140	13.86
AABBCC	0.1064	AABBCC	0 20 40 70 100 130	9.65
AABCCC	0.0794	AACCBC	0 20 40 70 100 130	12.01
ABCCCC	0.0675	ACCCCB	0 20 50 80 110 140	20.48
AABBBC	0.0634	AABBBC	0 20 40 70 100 130	7.57
BBBCCC	0.0611	BBCCCB	0 20 50 80 110 140	23.44
BBCCCC	0.0513	BCCCCB	0 20 50 80 110 140	27.40
ABBBBC	0.0481	ACBBBB	0 20 50 80 110 140	11.24
AAABCC	0.0467	AABCAC	0 20 40 70 110 130	6.95
AAABBC	0.0417	ABBCAA	0 20 50 80 120 140	5.16
BBBBCC	0.0409	BBCCBB	0 20 50 80 110 140	19.89
BCCCCC	0.0230	BCCCCC	0 20 50 80 110 140	31.88
AACCCC	0.0222	AACCCC	0 20 40 70 100 130	14.84
AAACCC	0.0174	AACCAC	0 20 40 70 110 130	8.82
ACCCCC	0.0151	ACCCCC	0 20 50 80 110 140	24.90
BBBBBC	0.0146	BBBCBB	0 20 50 80 110 140	16.82
AABBBB	0.0142	ABBBBA	0 20 50 80 110 140	5.91
AAAABC	0.0137	ACBAAA	0 20 60 90 120 140	3.53
AAABBB	0.0124	BBBAAA	0 30 60 90 120 140	4.21
ABBBBB	0.0086	ABBBBB	0 20 50 80 110 140	9.06
AAAACC	0.0077	ACACAA	0 20 60 80 120 140	4.65
AAAABB	0.0061	BABAAA	0 30 60 90 120 140	2.79
CCCCCC	0.0043	CCCCCC	0 20 50 80 110 140	38.60
BBBBBB	0.0022	BBBBBB	0 20 50 80 110 140	14.18
AAAAAC	0.0018	AACAAA	0 20 50 90 120 140	2.39
AAAAAB	0.0016	BAAAAA	0 30 60 90 110 140	1.79
AAAAAA	0.0002	AAAAAA	0 20 50 80 110 140	0.96

expected relative frequencies of A, B, and C requiring appointments. The scheduler would choose the schedule for a day from this subset on the basis of the relative frequency of open appointments on days for which the sequence is already chosen. For example, if the open A slots exceed 0.237 of the total open slots for the days on which the sequence is already fixed, then the decision should be to fix the next undecided day at some sequence with few slots of class A.

Assume for the moment that short-term variations in the relative frequency of appointment demands by the three customer classes can be handled. The problem is then: What relative frequencies should the above sequences be selected in to attain the required long-term relative frequencies of A, B, and C so that the expected cost is minimized? This problem is in the form of a linear program, in which the relative probabilities are the decision variables. Four constraints are imposed: all decision variables are nonnegative; the decision variables must sum to 1.0; and the relative expected frequencies of A, B, and C must be in the ratios of 0.237 : 0.360 : 0.403. The solution has an expected cost of 13.07 and requires the use of only the three sequences shown in Table 21. This strategy of only permitting these three sequences

Table 21. Optimal sequence set for the medical study. Because of variations in the proportion of requests from each class, this set can only support the stated goals 79% of the time.

sequence	schedule	cost	theoretical frequency	observed frequency
ACBCBB	0 20 50 80 110 140	13.86	0.5784	0.648
AACCBC	0 20 40 70 100 130	12.01	0.4200	0.344
ACBBBB	0 20 50 80 110 140	11.24	0.0012	0.008

allows variations in the relative frequencies of $A \in [\frac{1}{6}, \frac{1}{3}], B \in [\frac{1}{6}, \frac{2}{3}], C \in [\frac{1}{6}, \frac{1}{2}]$. The actual variations encountered in the relative frequencies of requesting customers may be great enough to cause two problems. The first is that some schedule slots go unfilled, for lack of customers of the correct type making requests in time to use the slots. The second is that a large number of customers of a single type making

requests in a small time period may cause an unacceptably long delay between a request for the appointment and the appointment itself (hereafter simply called the *delay*).

To approach these problems, a simulation of the appointment system was developed using FORTRAN 90. A simple set of rules determined the choice of combination for the next schedule. In a series of trials of 100 days of operation, the cost of the above strategy was 13.52 with a standard deviation of 0.02, slightly higher than the theoretical cost. As seen in Table 21, the frequencies of the three combinations were not close to the theoretical values, either. The discrepancies stem from three main sources. The first is that no value was placed on unfilled appointments. There were on average 8.0, 11.4, and 5.5 unfilled slots of types A, B, and C, respectively. These were not accounted for theoretically. A second source of discrepancy stems from the fact that theoretical solution is steady-state. This simulation was run for only 100 days, retaining the initial scheduling period in the tally. When the simulation was run for 900 days, the average cost dropped to 12.8. Last, the simulation's rules for selecting the combination for a given day were not optimal. Given these differences, the agreement between theory and simulation is good.

The doctor desired that the maximum delay between request and appointment be one week (5 scheduling periods). The simulation revealed that an average of 14.0% of the customers exceeded this limit, with the longest delay observed being 11 scheduling periods. Assuming this delay is deemed excessive, a reasonable solution might be to retain the three combinations in the solution above, but also to employ the sequences BBBAAA (cost of 4.21) and CCCCCC (cost of 38.6) when necessary. Table 22 summarizes this strategy. Using a similar rule base to that above, and applying the same set of random number streams, the simulation yielded a cost of 13.9, with a standard deviation of 0.06. An average of 2% of the customers suffered delays of over a week, with the maximum delay observed being 7 scheduling periods. Thus, the goal of reducing delays can be nearly met with the 5-combination

strategy, with only a 3% increase in cost over the 3-combination strategy. Further reductions in delays are possible with the inclusion of a few more possible combinations. What is the expected cost of the current approach, which entails

Table 22. The 98% solution for the medical study. Extending the number of feasible combinations to 5 increases the cost 3% but reduces the number of unacceptable delays from 14% to 2%. The frequencies noted are the relative frequencies observed over ten runs of the simulation of a 100-day schedule.

sequence	schedule	cost	frequency
ACBCBB	0 20 50 80 110 140	13.86	0.547
AACCBC	0 20 40 70 100 130	12.01	0.314
ACBBBB	0 20 50 80 110 140	11.24	0.020
BBBAAA	0 30 60 90 120 140	4.21	0.066
CCCCCC	0 20 50 80 110 140	38.6	0.053

scheduling customers into preset slots of approximately 20 minutes without regard to the classes defined here? A good estimate of the cost of this current system would require determining the cost of each permutation of the 22 possible combinations, for each of the 16 different schedules observed in the data. This requires excessive calculation, and it was judged that sufficient accuracy could be obtained for the purpose by considering only the 3 most frequently observed 6-customer schedules and the permutations of only the 8 most probable combinations of classes. Assume the schedule is one of [0 20 40 70 90 110 , [0 40 60 110 130 150], or [0 60 80 110 130 150]. These were the most common schedules observed, each accounting for 15% of the total. Assume the combination for each day is one of ABBCCC, BBCCCC, BBBCCC, AABBBC, ABCCCC, AABCCC, AABBCC, ABBBCC, or ABBCCC. These 8 combinations will account for 70% of the total, if the relative frequencies of classes A, B, and C are those observed in the data set.

If the sequence of customers is chosen at random from the above subset of 8, the expected costs of the three schedules are found to be 36.4, 49.5, and 60.5,

respectively, for an average of 48.8. This is taken here as the current daily cost of operation.

In real terms, this suggests that the equivalent of 48.8 minutes of productive time is lost, either through the doctor having to spend extra time at the end of the scheduling period (time that is weighted by a factor of 3) or through patients having to wait. The potential improvement attained by sequence and schedule optimization is thus 67%.

While the patients in this particular study were all military retirees, it is instructive to obtain a rough estimate of savings if the proposed scheme were implemented, if the patients had been active-duty U.S. Air Force servicemembers. The average hourly cost to the government of a servicemember's time is \$21.13,[2] and the doctor sees patients for about 260 days each year. The yearly savings for that practice alone is thus $\$21.13 \cdot 48.8/60 \cdot 260 \cdot 0.67 \approx \$3000$. This particular practice had a comparatively low waiting time per patient to begin with, so this is a conservative estimate of the potential savings that could be accrued from implementing such a scheme for each of the other outpatient practices in this hospital. It should be emphasized that this is the savings to the government as a whole; since the hospital would benefit financially from only the reduction in doctor overtime, and since even that savings might not have direct financial impact, there might be less motivation to adopt the new scheme.

It is also instructive to consider the relative improvements when the schedule and sequence are optimized. Suppose some better choice of schedule is imposed each day while still ignoring the class of the patient. Here, [0 20 50 80 100 130] was chosen as a reasonable schedule. It differs from each of the optimal schedules in Table 20 by at most 10 minutes for each customer arrival. The expected daily cost

[2]This average hourly cost for a USAF servicemember is derived from the pay rates in Air Force Instruction 65-503, Attachment 20, for FY96 (fiscal year 1996), dated 22 March 1996. These pay rates were prorated using the force strengths for FY96 cited in tables in Air Force Magazine, May 1996, p41. This calculation includes the cost to the Department of Defense of most benefits.

under this scheme is 22.2. This represents an improvement over current operations of 55%. This improvement is commensurate with that found by researchers who recommended appointment schedule (but not sequence) optimization for specific operations on the basis of simulations, such as Soriano's 50% improvement [149] or Glendenning's 25% improvement [51].

Compare this improvement to that attained when another (less realistic) policy is chosen. Suppose the sequences and schedules for each day are chosen at random from the above possibilities, but once chosen, the customers are ordered optimally. The costs for the three combinations become 27.8, 34.2, and 49.0, respectively, for an average of 37.0, a decrease of 24% from the current cost. It has been the case in each of a small series of tests that the improvement attained by optimizing the schedule is larger than that attained by optimizing the sequence of customers.

E.4 Outcome

The proposed policy was not implemented by the clinic, since this was only a feasibility study and was specific to a single physician, who had already moved to another job. The general approach seems highly promising as a way of reducing patient waiting time and doctor overtime. However, the proposal to expand the study and implement results on a small sample met with some reluctance from the clinic for several reasons:

- The clinic had just lost two of its three schedulers, due to personnel cuts, and it was deemed a poor time to make any changes that would complicate the scheduling process.
- Another recent study put the current average waiting time in the clinic at 9 minutes per patient, and this was deemed sufficiently low; reducing waiting time was not considered a high priority.

- The doctor with whom the study was performed has a very specialized practice, and it was not clear that similar benefits would accrue to other practices. In particular, the patient load was much heavier for other doctors.
- The minimum acceptable lattice size is 10 minutes. Because that is so large relative to the mean service time, it was thought that a modification of the scheduling protocol would not achieve much improvement. It was suggested that a surgery or other practice with longer appointments might accrue greater benefit.
- There was already a patient classification scheme in place, used for reasons other than scheduling effectiveness. Any scheme that forced other patient orderings would not be acceptable.

Some of these objections are easily dismissed. For instance, in the preliminary study above, if decreasing waiting time is not a priority, waiting time and overtime could be held at the same level and a seventh patient added to the schedule, without increasing the cost. Alternatively, the schedule horizon could be shortened by 25 minutes without increasing the cost. In response to concerns over the forced coarseness of the lattice, this did not prevent substantial improvement in the preliminary study.

It was proposed that, even though sequence optimization is objectionable, an implementation of a different schedule could still reduce waiting time substantially. In the above case, such a strategy led to 55% improvement. This point is being considered, and such a scheme could be adopted once personnel pressures ease.

Other objections clearly cannot be argued away by those outside the profession. The additional burden on administrative staff of the new protocol and the sacrosanctity of the current patient classification and ordering system precluded any further testing of the proposed sequencing optimization in this clinic.

In summary, a preliminary examination of a particular medical scheduling protocol indicated that the cost of operation could be reduced by a factor of 3, where the cost is defined as the sum of the patient waiting times and three times the doctor's overtime. No further resources are needed to achieve this improvement. Changes to the scheduling procedure include the scheduler questioning the patient as to whether his or her ailments have been attended to before by this doctor and the restriction of certain classes of patients to certain scheduling slots.

Further work on scheduling and sequencing patients in the clinic has not been pursued further (although the hospital has expressed interest in the use of the patient classification scheme above as a way of reducing service time variance, in a manner similar to the scheme proposed by Davis and Reed in the context of operating room scheduling [31]). The main reasons for reluctance, in the eyes of this researcher, are personnel cuts in the scheduling area and a general suspicion of the practicality of such nonintuitive results.

However, this preliminary study served its purpose well. It shows the potential of appointment schedule/sequence/combination optimization is substantial and provides a model for future studies which can be employed or improved on.

Appendix F. Matching Moments with Coxian Distributions

This appendix examines approaches to obtaining parameters for a Coxian distribution with a given set of moments. In the process, several apparently new results are obtained:

1. Necessary and sufficient bounds on feasible moments of Coxian-2 distributions,
2. A convenient framework for examining moment bounds,
3. An analysis of the feasible 3-moment space of the distribution defined by a Coxian phase appended to an Erlang distribution,
4. A recursive approach to determining moments, and
5. A graphical approach to determining equivalent representations of a phase-type distribution

It has been shown by others that phase-type distributions, in particular Coxian distributions, can provide arbitrarily accurate representations of any desired distribution with positive support (*i.e.*, for which each negative value has zero probability) [46, 73]. It remains to show methods of obtaining parameters (number of stages, rates, and transition probabilities) of the phase-type approximation desired. Usually, one is given empirical data representing the distribution. In this case, the EMPHT programs developed by Häggström, Asmussen, and Nerman [58] or the EM and ME programs developed by Asmussen, Nerman, and Olsson [5, 6] can provide the required parameters for a Coxian approximation. Alternatively, tools such as Johnson's MEFIT (Mixed Erlang Fit) can be used to fit a mixture of Erlang distributions, which can then be transformed to a Coxian distribution, as will be shown [72, 77]. These and similar tools are surveyed by Lang [89, 90].

In some situations, raw empirical data are unavailable, and only several moments are provided. This is often the case when one is trying to reproduce or expand

on others' earlier results. Moment matching is also a convenient tool for examining the sensitivity of measures of merit to higher moments of the distribution.

Theoretically, moment-matching may provide an accurate representation of the distribution. Let ϕ_k be the k^{th} noncentral moment. Cramér proved that if the series $\sum_{k=0}^{\infty} \phi_k c^r/k!$ is absolutely convergent for some positive value of c, then the distribution is uniquely determined by all of its moments [28]. Wilks proved that the distribution is uniquely determined by its moments if the support (the set of points for which the probability is nonzero) is bounded [170]. Practically, however, it is impossible to determine whether Cramér's condition holds, and Wilks's condition may not hold for service distributions of interest here. Whenever possible, one should match the CDF of a distribution rather than its moments.

The major goal of this appendix is to examine possible approaches to determining a phase-type distribution in which the first three moments match those of a given distribution. The reason for this choice is concern over whether the third moment of the service distribution strongly affects the optimal cost or schedule of an appointment system. While typical measures of merit in steady-state queueing systems are relatively insensitive to service distribution moments higher than the second, some researchers have noted sensitivity to the third moment when the coefficient of variation is greater than one [3, 169].

Once tools for matching moments of a given phase-type distribution are developed, they can be used to identify situations in which an appointment system's optimal cost and schedule may be sensitive to variations in the third moment. If the third moment proves to be inconsequential, fewer phases should be necessary to represent a given distribution, simplifying computation.

Tools such as MEFIT and Schmickler's MEDA (Mixed Erlang Distributions for Approximation) are already available to fit mixtures of Erlang distributions to the first three moments [138]. However, Johnson and Taaffe prove that a mixture of Erlang distributions is not the most parsimonious representation of a given moment

set, in terms of number of phases needed [73]. The parsimony issue is examined in more detail below.

F.1 Moment Space Coordinate System

Before examining the problem further, a convenient coordinate system will be described for exploring the moment space, which consists of all possible combinations of the first three moments. Unlike previous characterizations of this moment space, the measures of second and third moments to be used here will be noncentral, as well as scaled. The coefficient of variation is defined by $c = \sigma/\phi_1^2$, where σ^2 is the variance. The quantity $\Phi_2 = (\phi_2/\phi_1^2) = c^2 + 1$ will be used as a convenient measure of the scaled second moment. Rather than skewness, $\Phi_3 = \phi_3/\phi_1^3$ will be used as a measure of the third moment. These two measures will provide what will be shown to be a natural coordinate system for the 3-moment space, in which many relationships may be represented more simply.

Given a feasible combination of Φ_2 and Φ_3, any positive value of ϕ_1 always can be attained, simply by redefining the time variable to be the product of the current time divided by the ratio of the desired mean to the current mean. Since this scaling does not change the values of the scaled noncentral moments, the question of feasibility is one of characterizing the moment space in only two dimensions. In particular, once the parameters are obtained to match a phase-type distribution to the desired second and third scaled noncentral moments, the first moment can be matched simply by dividing each phase rate by the above ratio.

F.2 General Feasibility

What regions of this moment space are feasible for distributions with support on $\Re^+$? First, it is trivially clear that the bounds $\phi_k \geq 0$ hold for $k \in \mathcal{Q}^+$. Now

consider the quantity

$$\phi_3 = E[t^3] = E[((t-\phi_1)+\phi_1)^3] = \phi_1^3 + E[(t-\phi_1)^3] + 3\phi_1\sigma^2 \qquad (46)$$

The domain constraints $t \geq 0$ and $\phi_1 > 0$ imply $E[(t-\phi_1)^3] \geq -\phi_1^3$. Then $\phi_3 \geq 3\phi_1\sigma^2$, or $\Phi_3 \geq 3(\Phi_2 - 1)$, creating a necessary bound. Further, the skewness, defined by $\gamma = E[(t-\phi_1)^3]/\sigma^3$, is usually known to be positive for distributions on $\Re^+$. If this is the case, a tighter bound can be formed. Since $E[(t-\phi_1)^3] > 0$, $E[t^3] > 3\phi_1\sigma^2$, which is equivalent to $\Phi_3 > 3\Phi_2 - 2$. This bound is necessary and sufficient for any distribution on $\Re^+$ with positive skewness. Here, a necessary bound is one that must be met by a set of moments for feasibility. A bound is considered sufficient if all points that meet that condition and, in addition, meet other feasibility bounds, are feasible. A necessary and sufficient bound is one that is tight – *i.e.,* it delineates the set of feasible and infeasible points.

A necessary and sufficient limit for feasibility was obtained by Whitt [169] using Tchebycheff systems theory: $\phi_3\phi_1 > \phi_2^2$. Johnson and Taaffe put this bound in the equivalent form: $\gamma > c - c^{-1}$ [73, 76]. In the moment space representation suggested above, it is equivalent to $\Phi_3 > \Phi_2^2$. This bound is always tighter than $\Phi_3 \geq 3(\Phi_2 - 1)$, but the positive skewness bound, $\Phi_3 \geq 3\Phi_2 - 2$, is tighter when $\Phi_2 < 2$ (*i.e.,* $c < 1$). These bounds are shown in Figure 28.

F.3 Obtaining Coxian-r Moments

Given these conditions for feasibility for general distributions, it is natural to consider similar bounds when the distribution is constrained to a particular form. The ultimate goal is to represent a given moment set with the simplest Coxian distribution possible. Here, the Coxian phase rates, μ_i, are limited to $\Re^+$, transition probabilities are required to be positive, and all probability mass is assumed to be concentrated in the first stage at $t = 0$. While Cox [25] did not restrict his phase

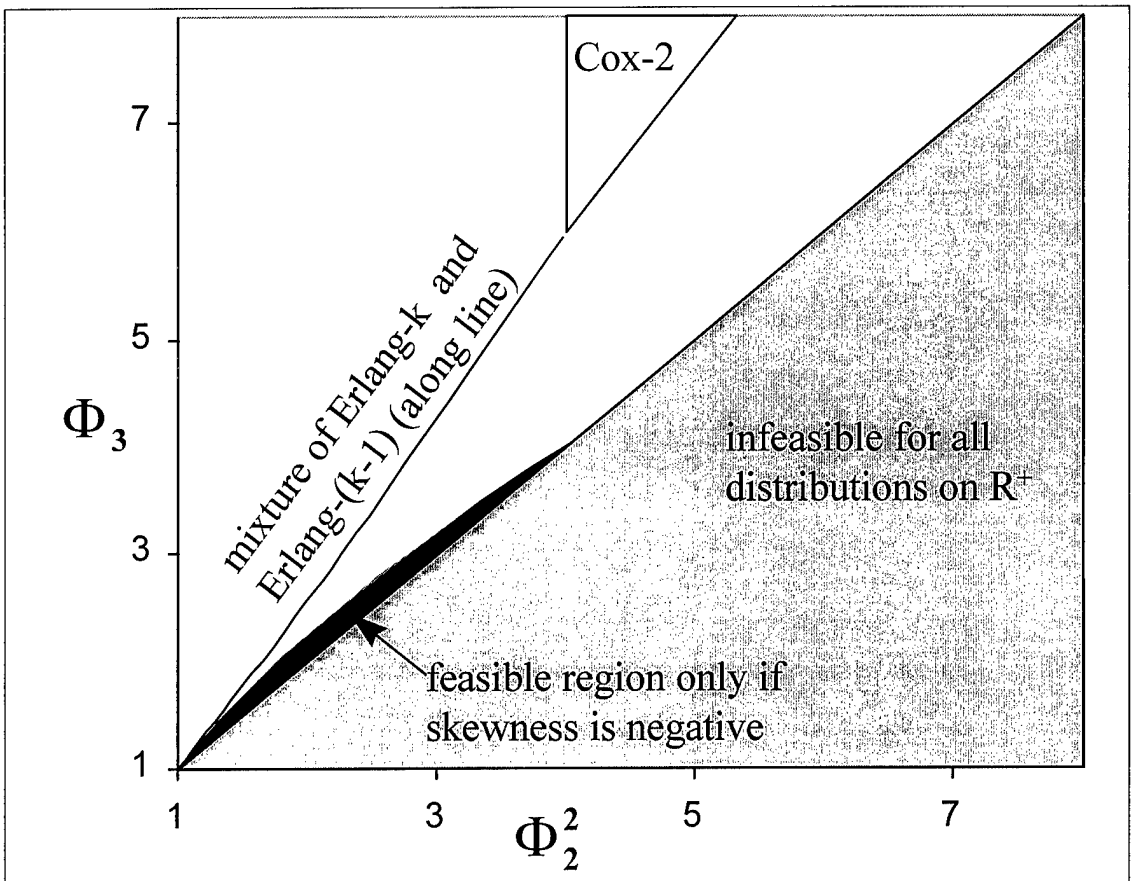


Figure 28. Feasible moment space

representation in these ways (indeed, did not even define it in terms of the physical representation in Figure 3), these restrictions are in common use today [3, 77, 117, 121].

In exploring the feasible moment space of Coxian distributions, it is necessary to obtain moments efficiently. While standard approaches such as differentiation of the Laplace transforms are available, the resulting expressions become unwieldy very quickly. For example, moment expressions for a Coxian-3 are:

$$\phi_1 = \frac{b_1\mu_1\mu_3 + b_1b_2\mu_1\mu_2 + \mu_2\mu_3}{\mu_1\mu_2\mu_3} \tag{47}$$

$$\phi_2 = \frac{2}{\mu_1\mu_2\mu_3}\left[\left(\frac{1}{\mu_1}+\frac{1}{\mu_2}+\frac{1}{\mu_3}\right)(b_1\mu_1\mu_3 + b_1b_2\mu_1\mu_2 + \mu_2\mu_3) - \mu_2 - \mu_3 - b_1\mu_1\right] \tag{48}$$

$$
\begin{aligned}
\phi_3 \;=\; & \frac{-2}{\mu_1\mu_2^2\mu_3^2}\begin{pmatrix} -\mu_3^2 + b_1b_2\mu_2^2 + b_1b_2\mu_1\mu_2 - 4\mu_2\mu_3 \\ -b_1\mu_1\mu_2 - \mu_2^2 + b_1\mu_3^2 + 2b_1b_2\mu_2\mu_3 \end{pmatrix} \\
& -\frac{8}{\mu_1\mu_2\mu_3}(\mu_2+\mu_3+b_1\mu_1)\left(\frac{1}{\mu_1}+\frac{1}{\mu_2}+\frac{1}{\mu_3}\right) \\
& +\frac{6}{\mu_1\mu_2\mu_3}(b_1\mu_1\mu_3 + b_1b_2\mu_1\mu_2 + \mu_2\mu_3)\left(\frac{1}{\mu_1^2}+\frac{1}{\mu_2^2}+\frac{1}{\mu_3^2}\right) \\
& +\frac{8}{\mu_1\mu_2\mu_3}(b_1\mu_1\mu_3 + b_1b_2\mu_1\mu_2 + \mu_2\mu_3)\left(\frac{1}{\mu_1\mu_2}+\frac{1}{\mu_1\mu_3}+\frac{1}{\mu_2\mu_3}\right) \qquad (49)
\end{aligned}
$$

Clearly, it will be oppressive to obtain moments via Laplace transforms or moment generating functions. One alternative is to use a matrix-geometric representation [117]:

$$\mu_k = (-1)^k k! T^{-k} \Psi_r^T \qquad (50)$$

where T is the state transition matrix and Ψ_r^T is a column vector of length r with all elements equal to unity, as discussed in Section 3.2. Another approach is to generate the moments recursively from the Laplace transforms. Consider the recursive representation of a Coxian-j in Figure 29.

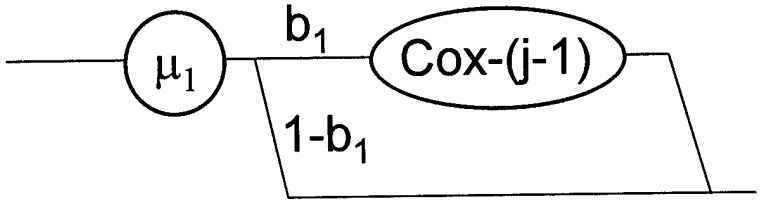


Figure 29. Recursive representation of a Coxian-j

Let $F_j(s)$ be the Laplace transform of a Coxian-j distribution. Let $F_j^{[k]}(s)$ be its k^{th} derivative with respect to s. Then $\phi_{j,k} = (-1)^k F_j^{[k]}(0)$ is the k^{th} moment of a Coxian-j. Note that $F_1(s) = \mu/s+\mu$, $F_1^{[k]}(s) = (-1)^k k!\mu/(s+\mu)^{k+1}$, and $\phi_{1,k} = k!/\mu^k$. Now suppose one wished to add a phase to the beginning of a Coxian-$(j-1)$ to create a Coxian-j, with rate μ and probability b. Then the following holds

for $j > 1$:

$$\begin{aligned}
F_j(s) &= (1-b)F_1(s) + bF_1(s)F_{j-1}(s) \\
F_j^{[k]}(s) &= (1-b)F_1^{[k]}(s) + b\sum_{i=0}^{k}\binom{k}{i}F_1^{[k-i]}(s)F_{j-1}^{[i]}(s) \\
&= F_1^{[k]}(s) + b\sum_{i=1}^{k}\binom{k}{i}F_1^{[k-i]}(s)F_{j-1}^{[i]}(s) \\
(-1)^k F_j^{[k]}(s)\Big|_{s=0} &= (-1)^k F_1^{[k]}(s)\Big|_{s=0} + b\sum_{i=0}^{k}(-1)^{2k-i}\frac{k!}{i!}\frac{1}{\mu^{k-i}}F_{j-1}^{[k]}(s)\Big|_{s=0} \\
\phi_{j,k} &= \frac{k!}{\mu^k} + k!b\sum_{i=1}^{k}\frac{\phi_{j-1,i}}{i!\mu^{k-i}} \qquad (51)
\end{aligned}$$

This recursive relation provides a simple alternative to matrix formulations when programming.

F.4 Coxian-2 Feasibility Bounds when $c > 1$

Consider a Coxian-2 distribution. Defining $w = \mu_2/\mu_1$, and defining $b = b_1$ as the transition probability from the first to second phase, the recursive equation above leads to

$$\phi_n = \frac{n!}{w^n\mu_1^n}\left(w^n + b\sum_{i=0}^{n-1}w^i\right) \qquad (52)$$

Algebraic manipulation of the first three moments leads to

$$\Phi_3 = \frac{3}{2}\Phi_2^2 + \frac{3w(\Phi_2 - 2)}{(w+b)^2} \qquad (53)$$

It is clear that if $\Phi_2 > 2$ (*i.e.*, if $c > 1$), a necessary condition for feasibility is $\Phi_3 > \frac{3}{2}\Phi_2^2$. To see this bound is sufficient, consider the following expression for Φ_2, obtained from the moment expressions:

$$\Phi_2 = 2(w^2 + wb + b)/(w+b)^2 \qquad (54)$$

In the limit as w approaches zero, Φ_3 approaches $\frac{3}{2}\Phi_2^2$ and Φ_2 approaches $2/b$, which by judicious choice of b can take on any value greater than 2. The bound $\Phi_3 \geq \frac{3}{2}\Phi_2^2$ is therefore tight when $\Phi_2 > 2$. No other bounds pertain when $\Phi_2 > 2$. These results were obtained by Altiok by a more involved argument [3]. This bound is depicted in Figure 28.

F.5 Equivalence of Phase-Type Distributions

Before proceeding with other bounds of the Coxian-2, Whitt's work on the hyperexponential distribution with two phases (H_2) should be noted. A hyperexponential distribution, H_r, is a mixture of r exponential phases. A mixture is defined by a set of distributions in parallel, with the distributions assigned mutually exclusive and collectively exhaustive branching probabilities. Whitt found that the feasible moment space for the H_2 is delineated by $\Phi_2 > 2$ and $\Phi_3 > \frac{3}{2}\Phi_2^2$, the same bounds Altiok later obtained [169]. Johnson and Taaffe observed that the reason for the equivalence of these bounds is that an H_2 is equivalent to a Coxian-2 with $\mu_1 \geq \mu_2$ [73].

This equivalence is most easily seen by applying Cumani's results [29]. He showed by a simple algebraic identity that any phase could be replaced by a mixture of that phase and one with a larger transition rate, without changing the Laplace transform of the phase-type distribution. Still letting $w = \mu_2/\mu_1$,

$$\frac{\mu_2}{s+\mu_2} = w\frac{\mu_1}{s+\mu_1} + (1-w)\frac{\mu_1\mu_2}{(s+\mu_1)(s+\mu_2)} \tag{55}$$

Figure 30 shows the transformation of an H_2 distribution into an equivalent Coxian-2. This transformation is reversible as long as $\mu_1 \geq \mu_2$, which is tantamount to requiring $c \geq 1$. This graphical approach to obtaining phase-type equivalence can be rigorously supported and is more intuitive than manipulating Laplace transforms.

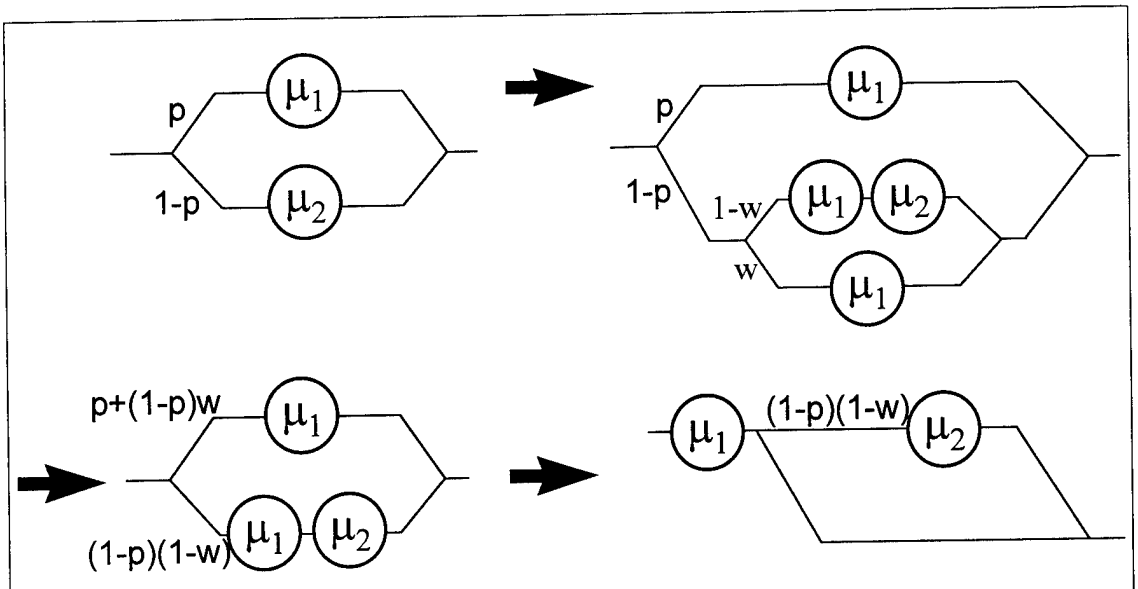


Figure 30. Transformation of H_2 to Coxian-2 using Cumani's result [29]

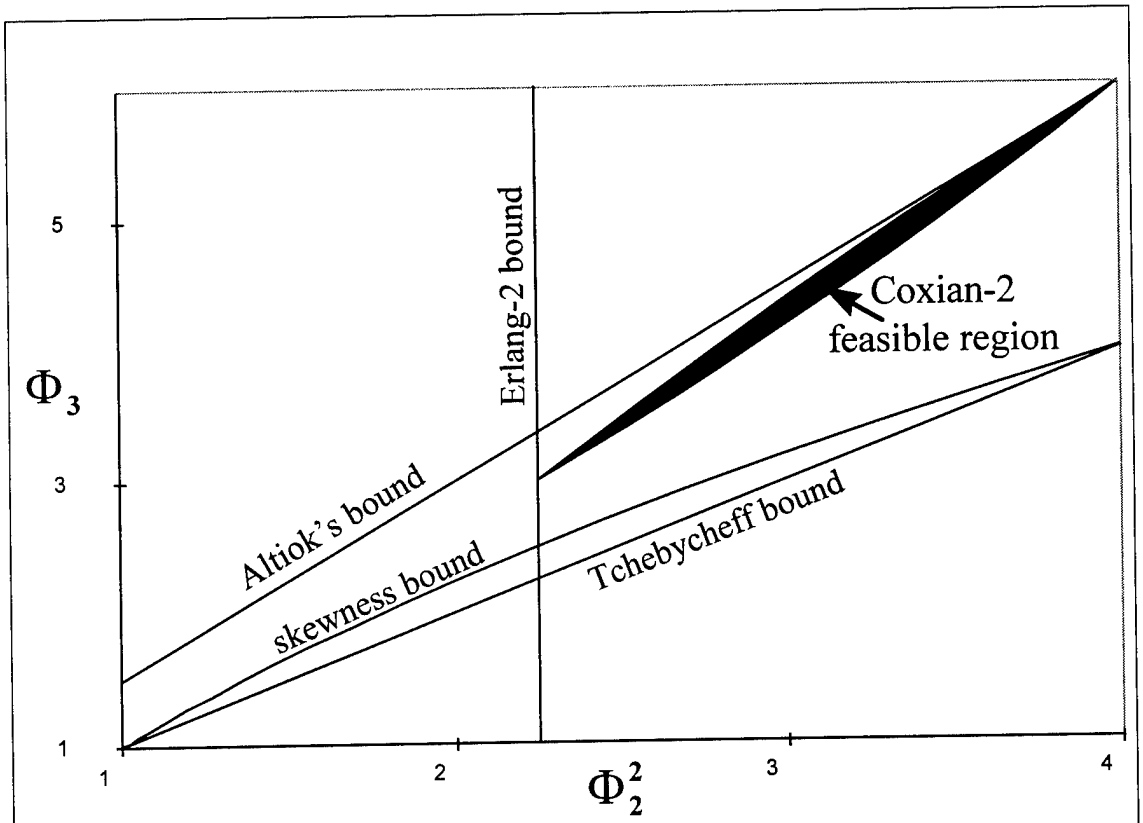


Figure 31. Coxian-2 moment space bounds with $\Phi_2 < 2$

F.6 Coxian-2 Feasibility Bounds when $c < 1$

If $\Phi_2 < 2$, then Equation (53) leads to the necessary bound $\Phi_3 < \frac{3}{2}\Phi_2^2$. This is marked on Figure 31 as Altiok's bound, although Altiok was not concerned with the case of $\Phi_2 < 2$.

By further manipulation of the moment expressions for a Coxian-2,

$$\Phi_3 = 6(\Phi_2 - 1) - \frac{3(1-b)(2-\Phi_2)}{w+b} \tag{56}$$

When $\Phi_2 < 2$, Equation (56) leads to the necessary upper bound $\Phi_3 \leq 6(\Phi_2 - 1)$. To see this bound is also sufficient, note that it is attained only when $\Phi_2 = 2$ or $b = 1$. If b is set to unity, $\Phi_2 = 2 - w/(w+1)^2$, from which it is clear that w can be chosen to obtain any desired value of $\Phi_2 \in [1.5, 2.0]$. Thus the bound $\Phi_3 \leq 6(\Phi_2 - 1)$ can be attained and cannot be surpassed; it is tight. This bound is depicted in Figure 31 as the upper bound of the Coxian-2 feasible region.

One necessary lower bound to the Coxian-2 with $\Phi_2 < 2$ is provided by the general feasibility condition, noted on Figure 31 as the Tchebycheff bound. It is also referred to by Johnson and Taaffe as the Bernoulli bound, presumably since it can only be attained by a generalized Bernoulli distribution [75]. This bound can be tightened by further algebraic manipulation of the moment expressions to obtain

$$\gamma = \frac{2(w^3 + 3b(b^2 - 3b + 3))}{c^3(w+b)^3} \tag{57}$$

Since the quadratic term is positive for all values of b, the skewness is always positive. Another necessary lower bound is thus the positive skewness bound found earlier: $\Phi_3 > 3\Phi_2 - 2$. This is marked in Figure 31 as the skewness bound.

A tight lower bound may be obtained by solving Equation (54) for b and substituting into Equation (56) to obtain

$$\Phi_3 = \frac{6\Phi_2^2(w^3 + 2w^2 - \Phi_2 w^2 + w + 1 + x(w^2 + w + 1))}{(w + x + 1)^3} \tag{58}$$

where $x = \sqrt{(w+1)^2 - 2\Phi_2}$. For a fixed Φ_2,

$$\frac{\partial \Phi_3}{\partial w} = \frac{6(w-1)\Phi_2^2(2 - \Phi_2)(2w^2 + w + 1 - \Phi_2 w^2 + wx + x)}{x(w + x + 1)^4} \tag{59}$$

This partial derivative disappears at $w = 1$. The last parenthetical term does not disappear for $\Phi_2 \in [1.5, 2]$. Since

$$\left.\frac{\partial^2 \Phi_3}{\partial w^2}\right|_{w=1} = \frac{6(2 - \Phi_2)\Phi_2^2(\sqrt{2}(5 - 3(\Phi_2 - 1)^{\frac{3}{2}}) + \sqrt{2 - \Phi_2}(8 - (\Phi_2)}{\sqrt{2 - \Phi_2}(2 + \sqrt{2(2 - \Phi_2)})^5} \tag{60}$$

is positive for $\Phi_2 \in [1.5, 2]$, $w = 1$ represents the minimum Φ_3 for a fixed Φ_2. Substitution of $w = 1$ leads to the two equations

$$\begin{aligned} \Phi_3 &= \frac{18(b+1)}{(b+1)^3} \\ \Phi_2 &= \frac{2(2b+1)}{(b+1)^2} \end{aligned} \tag{61}$$

which combine to give a tight lower bound:

$$\Phi_3 = 9\Phi_2 - 10 + 3\sqrt{2}\,(1 - \Phi_2)^{\frac{3}{2}} \tag{62}$$

This bound is equally cumbersome when defined with respect to c:

$$\Phi_3 = 3\left(3c^2 - 1 + \sqrt{2(1 - c^2)^3}\right) \tag{63}$$

It is depicted in Figure 31 as the lower bound of the area marked "Coxian-2 feasible region".

Aldous and Shepp proved that the least variability in a phase-type distribution is attained by an Erlang distribution [2]. Therefore, the bound $\Phi_2 \geq 1.5$ ($c^2 \geq 0.5$) also applies to the Coxian-2. This is marked in Figure 31 as the Erlang-2 bound. As can be seen, it is rendered redundant by the combination of the above tight upper and lower bounds, which meet at $\Phi_2 = 1.5$ and at $\Phi_2 = 2$ ($c^2 = 0.5$ and $c^2 = 1$).

To summarize the Coxian-2 moment space, if $c \geq 1$, the bound $\Phi_3 > \frac{3}{2}\Phi_2^2$ is necessary and sufficient. If $\Phi_2 < 2$, the two bounds $\Phi_3 \leq 6(\Phi_2 - 1)$ and $\Phi_3 \geq 3\left(3\Phi_2 - 4 + \sqrt{2(2-\Phi_2)^3}\right)$ are necessary and sufficient.

The author verified these bounds by means of nonlinear programs. A Newton search using calculated second derivatives was applied, as was a conjugate gradient search. The objective function was a weighted combination of the squared error in Φ_2 from the desired value and the value of Φ_3. The weight on the error in Φ_2 was set very large, so that the search would stay very close to the desired Φ_2. The weight on Φ_3 was set to ± 1, depending on whether the minimum or maximum value was desired. Several starting points were applied to each problem.

The feasible area obtained in this way appears to be precisely that depicted in Figure 31. Attempts to match moment pairs lying just beyond the theoretical bounds met with failure; the solutions thus obtained lay very close to, but within, the bounds in every trial. This supports the bounds proven above.

F.7 Matching Two Moments

If only the first two moments are desired, the Coxian-2 is an adequate model when $\Phi_2 > 1.5$. When $\Phi_2 = r + 1/r$, with r integral, an Erlang-r suffices. When $r + 1/r < \Phi_2 < r/r - 1$, a mixture of an Erlang-r and Erlang-$(r-1)$ distributions suffice. In the pure Erlang case, $\Phi_3 = (2c+1)(c+1) = 2\Phi_2 + 3\sqrt{\Phi_2 - 1} - 1$. In the Erlang mixture case, the relation is more complicated, but it is approximated

quite accurately with the same expression. The resulting curve in 3-moment space is depicted in Figure 28.

F.8 Matching Three Moments

Using a Coxian-2 to match three moments is problematic; much of the 3-moment space is unobtainable via a Coxian-2, bounds had not been determined until now for $\Phi_2 < 2$, and no convenient algorithm exists for obtaining the Coxian parameters from the moments over much of the moment space. Hence, researchers have turned to other phase-type distributions to match three moments. Johnson and Taaffe proposed the use of mixtures of two Erlang-r distributions with distinct phase rates [73]. They showed that, for sufficiently large r, any set of feasible first three moments is reachable (except for degenerate cases that are obtainable only by concentrating the probability mass at one or two points). They also provided analytical formulas for determining the parameters required for matching the first three moments.

Johnson and Taaffe showed the number of phases must meet the conditions

$$\begin{aligned} c &\geq \frac{1}{\sqrt{r}} \\ \gamma &\geq \frac{1}{1+r}\left[c^{-3} + (1-r)c^{-1} + (2+r)c\right] \end{aligned} \tag{64}$$

The expressions for these conditions are substantially simpler when expressed in the (Φ_3, Φ_2^2) coordinate system (Figure 32):

$$\begin{aligned} \Phi_2 &\geq \frac{r+1}{r} \\ \Phi_3 &\geq \frac{r+2}{r+1}\Phi_2^2 \end{aligned} \tag{65}$$

The minimal value of r necessary to meet these equations is

$$r_{min} = \text{int}\left(\max\left[1, \frac{1}{\Phi_2 - 1}, \frac{\Phi_2^2}{\Phi_3 - \Phi_2^2}\right]\right) \tag{66}$$

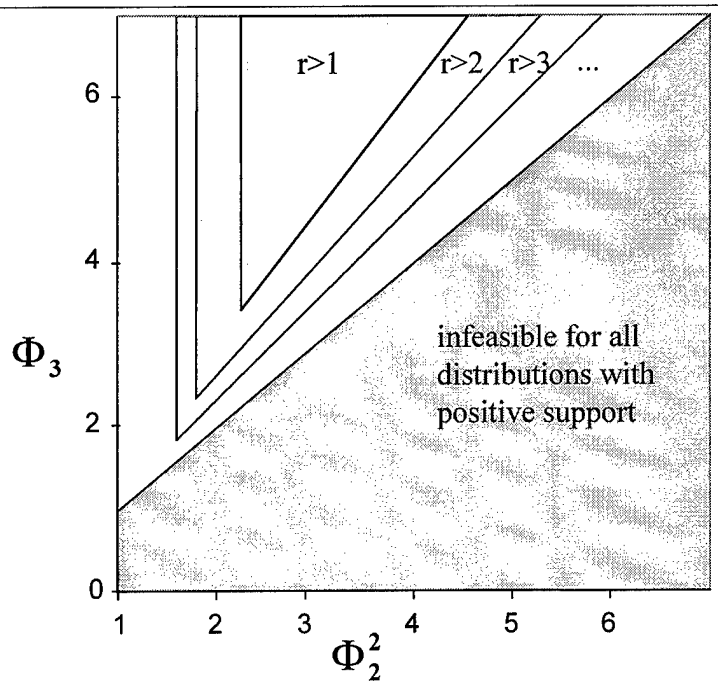


Figure 32. Feasible 3-moment space for a mixture of two Erlang-r distributions [73]. It will be shown in Section F.9 that this is also the sufficient space of a Cox-plus-Erlang-(r) distribution.

A mixture of an Erlang-$r(\mu_\alpha)$ and Erlang-$r(\mu_\beta)$, $\mu_\alpha \geq \mu_\beta$, is equivalent to at least one Coxian-$2r$ [29]. Further, there is only one Coxian-$2r$ with $\mu_1 \geq \mu_2 \ldots \geq \mu_{2r}$ that represents the distribution [121]. (Cumani's result shows directly that there are an infinite number of Coxian representations if the phase rates are not required to be ordered.) This unique distribution is depicted in Figure 33. It is a Coxian-$2r$ with parameters obtained by Cumani's relation as $\mu_1 = \mu_2 = \ldots = \mu_r = \mu_\alpha$, $\mu_{r+1} = \mu_{r+2} = \ldots = \mu_{2r} = \mu_\beta$, $b_1 = b_2 = \ldots = b_{r-1} = 1$, $b_r = 1 - p - w^r$, and $b_j = 1 - \binom{r}{j-r}(1-p)w^{j-r}(1-w)^{2r-j}$ for $j = r+1, r+2, \ldots 2r$. Again, $w = \mu_2/\mu_1$ and $\mu_1 \geq \mu_2$. Thus, the approach of Johnson and Taaffe leads to an analytical

procedure to obtain the parameters of a Coxian-$2r$ distribution whose first three moments match any feasible set.

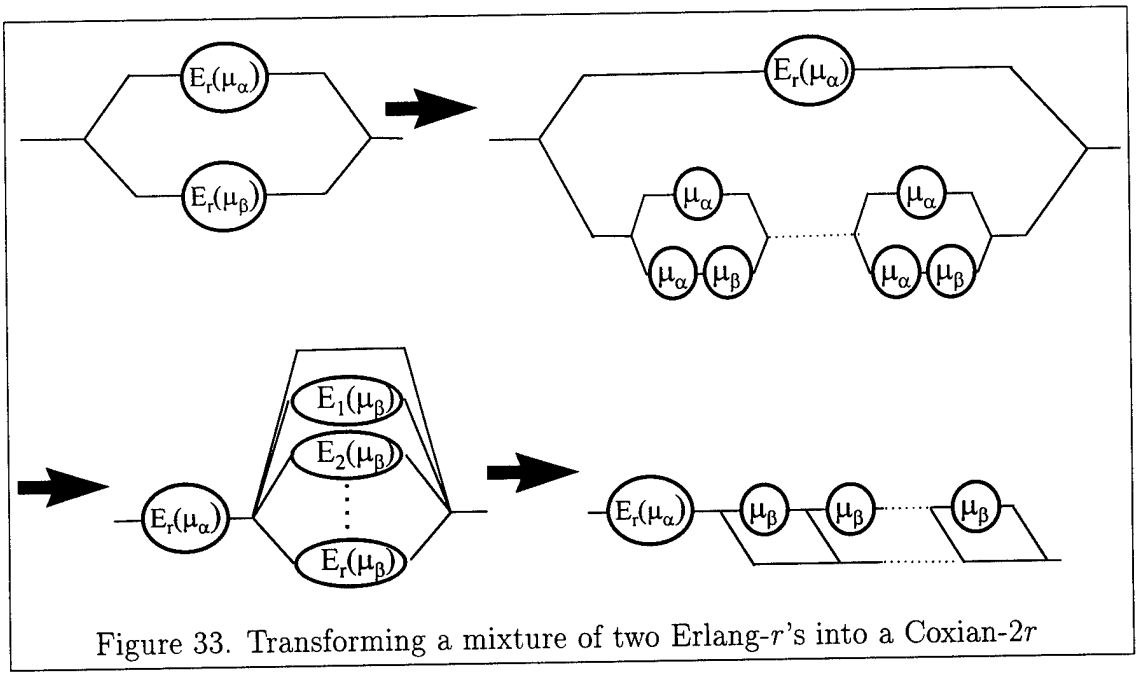


Figure 33. Transforming a mixture of two Erlang-r's into a Coxian-$2r$

The mixture of two Erlang-r's in Figure 33 has precisely the same Laplace transform as its Coxian-$2r$ counterpart, so all moments of the two match. If all that is required is matching the first three moments, substantially fewer phases are needed. Johnson and Taaffe proved that the feasible 3-moment space of a mixture of two Erlang-r's is precisely the same as the feasible 3-moment space of a mixture of an Erlang-r and an exponential [77]. This in turn is precisely equivalent to either the Coxian-$(r+1)$ in Figure 34 or that in Figure 35, depending on the ratio of the phase rates. Note that $w = \mu_\alpha/\mu_\beta$ in the first case and $w = \mu_\beta/\mu_\alpha$ in the second, so that it is always true that $w \in [0,1]$.

For the case in which $\mu_\alpha \geq \mu_\beta$, the b_j are easily computed and are shown in the figure. For the case in which $\mu_\alpha \leq \mu_\beta$, it can be proved recursively that

$$\begin{aligned} b_1 &= pw \\ b_j &= \frac{p(1-w)^j + 1 - p}{p(1-w)^{j-1} + 1 - p} \qquad \text{for } 2 \leq j \leq r-3 \end{aligned} \tag{67}$$

$$b_{r-2} = 1 - p + \frac{p(1-w)^j + 1 - p}{p(1-w)^{j-1} + 1 - p}$$

$$b_{r-1} = \frac{p(1-w)^r}{1 - p + p(1-w)^r w}$$

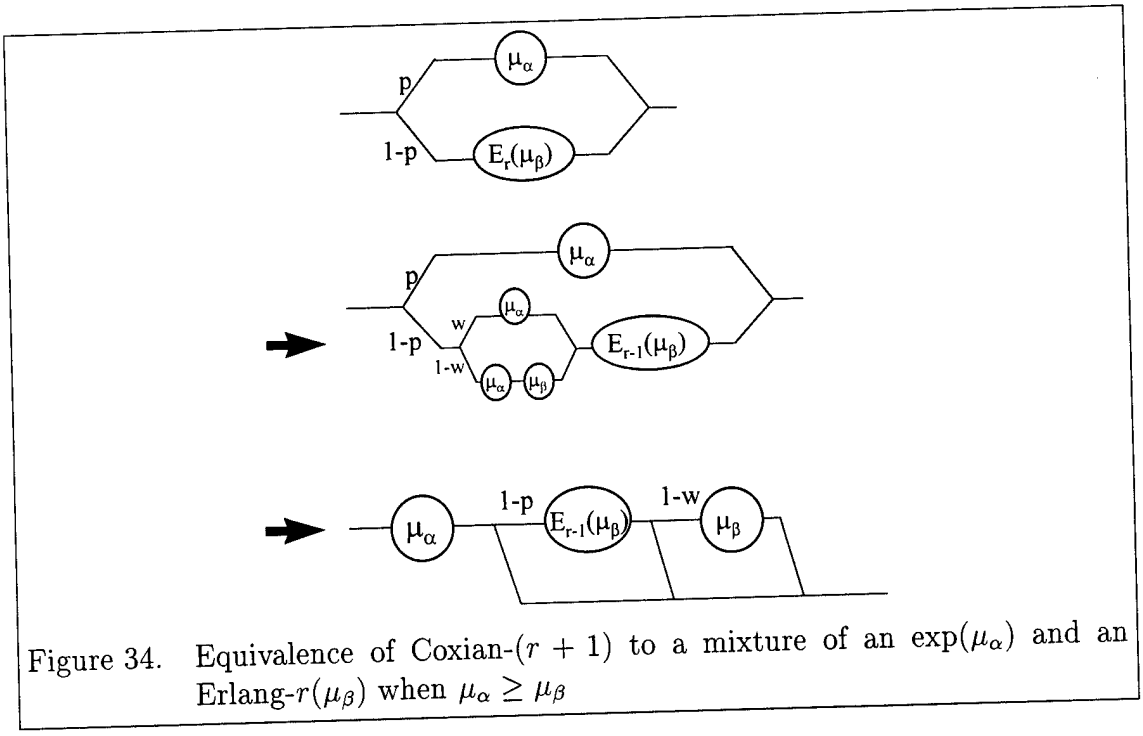


Figure 34. Equivalence of Coxian-$(r+1)$ to a mixture of an $\exp(\mu_\alpha)$ and an Erlang-$r(\mu_\beta)$ when $\mu_\alpha \geq \mu_\beta$

The above argument shows that the Coxian-$(r+1)$ is sufficient to model the first three moments of any distribution, provided Equations 66 are met. It does not, however, provide a method of obtaining the coefficients for the Coxian-$(r+1)$. If analytical determination is required, it is necessary to resort to the Coxian-$2r$. This author's attempts to solve the set of algebraic equations resulting from matching moments with the Coxian-$(r+1)$ have failed. If analytical expressions are not required, a nonlinear program (NLP) can be used to invert the moment expressions and obtain the coefficients.

F.9 Matching Three Moments with a Cox-Plus-Erlang-r Distribution

For any NLP, efficiency will degrade rapidly as the number of estimated parameters increases, so it is essential to consider ways of representing the 3-moment space

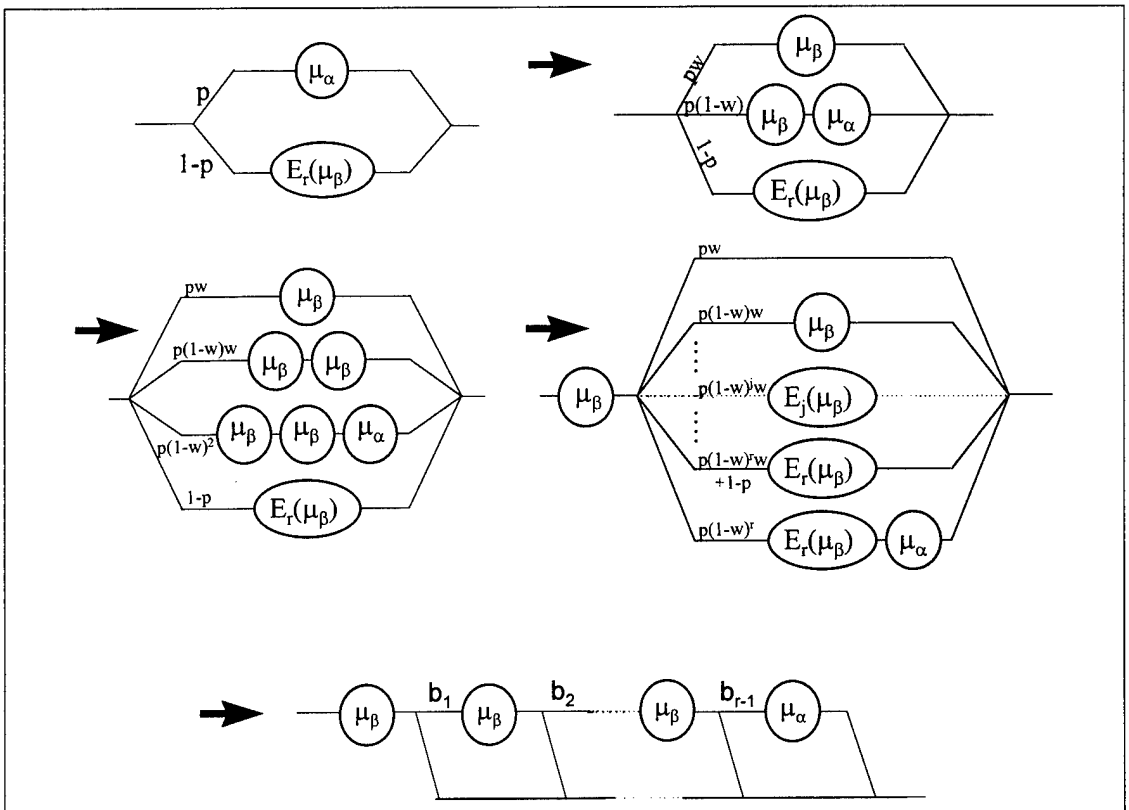


Figure 35. Equivalence of Coxian-$(r+1)$ to a mixture of an $\exp(\mu_\alpha)$ and an Erlang-$r(\mu_\beta)$ when $\mu_\alpha \leq \mu_\beta$

that reduce the number of parameters to be estimated. In addition, if a phase-type representation is desired, it is usually desirable to minimize the number of phases required. The most efficient choices discussed above are twofold. First, one can represent the moments with a mixture of an Erlang-r and an exponential distribution, then transform it to a Coxian-$(r+1)$ distribution, in which case two parameters must be estimated by means of an NLP. Second, one can represent the moments with a mixture of two Erlang-r distributions, in which case two parameters may be analytically determined. This section proposes a more efficient model, if $c > 1$.

Since only two of the $2r$ parameters of the Coxian-$(r+1)$ in Figures 34 and 35 are independent, it might seem that one could match three moments with fewer than $r+1$ phases, but experiments attempting to do so have not been successful. An NLP was constructed to determine the feasible 3-moment space of Coxian-$(r+1)$ distributions. For a variety of values of r, points very close to the conjectured boundaries were observed, but no points violating those boundaries were obtained. This suggests that these conjectured bounds are tight and that the full $r+1$ phases are necessary for 3-moment matching. Similar experiments have been performed by Johnson and Taaffe [75].

In their NLP model, Johnson and Taaffe allowed all parameters of the Coxian-$(r+1)$ to vary [75]. They found that, for points close to the boundary, their NLP solution strongly approximated two Coxian stages followed by an Erlang-$(r-1)$. (This is in contradistinction to the four-parameter Coxian distributions in Figures 33 and 34, which are proved to be capable of matching three moments.) Johnson's and Taaffe's result has been reproduced in this effort. It may be of some use when moment points are close to the boundary, since the surface formed when restricting the parameters to the form in Figures 34 and 35 creates local minima that impede progress of an NLP. The Cox-Cox-Erlang form does not exhibit local minima close to the boundary, which may be the reason such a solution is so often obtained when searching for a moment set in this region and allowing all parameters to vary.

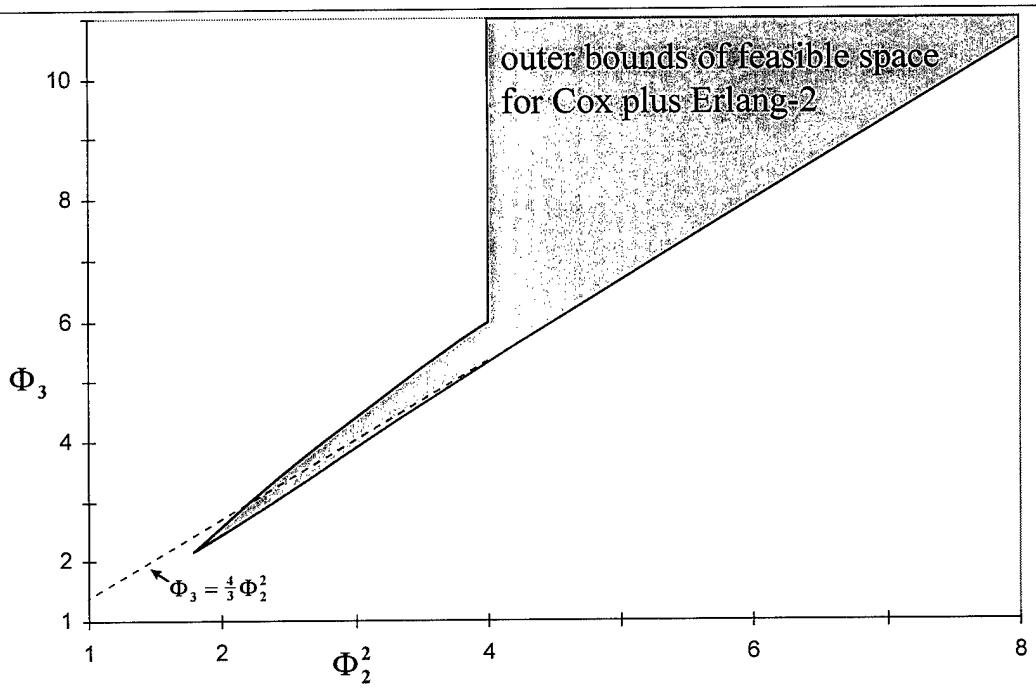


Figure 36. Moment space limits of a Coxian phase added to an Erlang-2 distribution. These are the empirically determined outer bounds of reachable distributions in 3-moment space.

Johnson's and Taaffe's observation does raise the question of feasible space when the model in Figure 34 is further constrained. For instance, empirical study led me to consider a single Coxian stage followed by an Erlang-r as an effective model when $\Phi_2 > 2$. This is a natural generalization of the Coxian-2, as suggested by the empirically determined bounds in Figure 36. The moment spaces of other Cox-plus-Erlang distributions exhibit the same general shape. This suggests the following theorem:

Theorem 20 *The Cox-plus-Erlang-r distribution can reach all 3-moment points for which* $\Phi_2 > 2$ *and* $\Phi_3 > \frac{r+2}{r+1}\Phi_2^2$.

Proof: The moment expressions for the Cox-plus-Erlang-r distribution are

$$\Phi_2 = \frac{2w^2 + 2brw + br^2 + br}{(w + br)^2} \tag{68}$$

$$\Phi_3 = \frac{6w^3 + 6brw^2 + 3br^2w + 3brw + br^3 + 3br^2 + 2br}{(w + br)^3} \tag{69}$$

As $w \to \infty$, $\Phi_2 \to 2$ in a continuous fashion. As $w \to 0$ and $b \to 0$, $\Phi_2 \to \infty$ continuously, so all values of Φ_2 greater than 2 can be reached. Suppose Φ_2 is fixed at one of these values and that r is also fixed. What are the possible values Φ_3 may take on?

Solving Equation (68) for b and substitution into Equation (69) yields an expression for Φ_3 in terms of w, r, and Φ_2:

$$\begin{aligned} \xi &= 2w + r + 1 \pm \sqrt{(2w + r + 1)^2 - 4w\Phi_2(r + 1)} \\ \Phi_3 &= \frac{4\Phi_2^2 \left[(r + 1)(r + 3w + 2)(\xi - 2w\Phi_2) + 6w^2\xi \right]}{\xi^3} \end{aligned} \tag{70}$$

Allowing $w \to 0$ yields

$$\begin{aligned} \lim_{w \to 0} \xi &= (r + 1) \pm (r + 1) \\ \lim_{w \to 0} \Phi_3 &= \frac{4\Phi_2^2(r + 1)(r + 2)}{\xi^2} \end{aligned} \tag{71}$$

and substitution of the two values of ξ yields

$$\lim_{w \to 0} \Phi_3 = \begin{cases} \infty & \xi = r + 1 \\ \frac{r+2}{r+1}\Phi_2^2 & \xi = -r - 1 \end{cases} \tag{72}$$

Thus, as $w \to 0$, it is shown that both the desired lower bound of Φ_3 is approached and also that Φ_3 is unbounded above, depending on the branch of ξ taken. If Φ_3 is continuous with respect to w for a fixed Φ_2, then all points between these bounds can be reached as well. The only threat to this continuity lies in the possible nonexistence of ξ. The situation may be seen in Figure 37. When $w < (r+1)/(4\Phi_2(r+1)-2) = \eta$, ξ is bifurcated, and each branch is continuous with respect to w over $w < \eta$. Further,

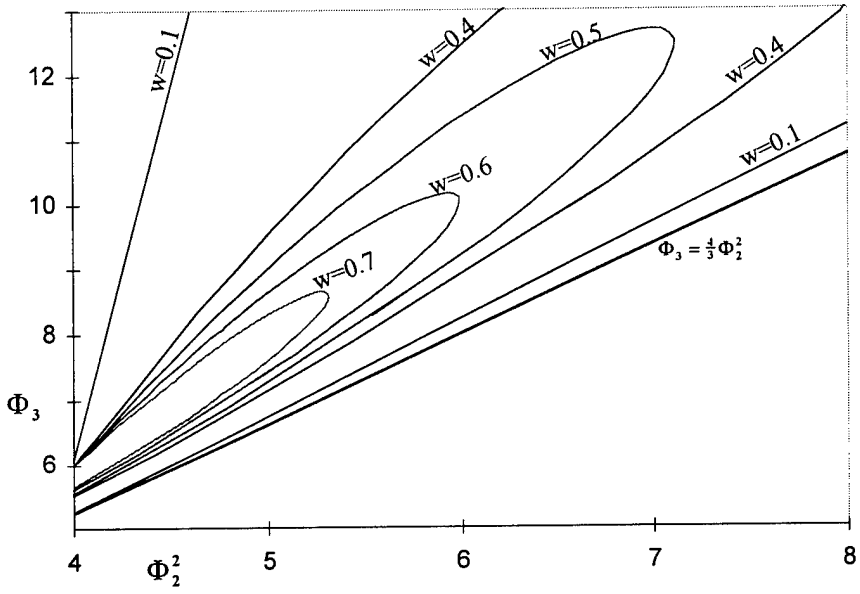


Figure 37. Constant-w contours for a Cox-plus-Erlang-2 as b is varied. All points for which $c > 1$ and that are reachable with a Cox-$(r+1)$ distribution are also reachable with a Cox-plus-Erlang-r distribution.

they both converge to the same value at $w = \eta$, so ξ is continuous with respect to w over $w \leq \eta$. Then Φ_3 is continuous over $w \leq \eta$ as well, and by the intermediate value theorem, it can attain any value in $(\frac{r+2}{r+1}\Phi_2^2, \infty)$, using some $w \in [0, \eta]$. ▪

In the special case of $w = 1$, the Cox-plus-Erlang-r is useful in matching just the first two moments when $1/(r-1) \leq c^2 \leq 1$. The distribution is then equivalent to a mixture of an exponential and an Erlang-$(r-1)$, which Tijms calls an $E_{1,r-1}$ distribution [154]. Closed expressions for the parameters as a function of the desired moments are easy to obtain in this case .

Theorem 20 shows that, when $c > 1$, the Cox-plus-Erlang-r distribution can reach any set of three moments that is (conjectured to be) reachable by a Coxian-$(r+1)$ distribution. Use of this restricted Coxian-$(r+1)$ provides a fast and parsimonious approach to matching any feasible three moment set when $c > 1$. Rather than a search over the $2r-1$ variables required in an unconstrained Coxian-r distribution,

the minimum value of r is selected analytically, using Equations (66). When $r = 1$, a Coxian-2 is required, and parameters can be determined analytically using Altiok's approach [3]. For $r > 1$, a Cox-plus-Erlang-r is required. A line search over w is performed, employing Equations (68), (69), and (71), after which the other parameters can be determined analytically. A Microsoft Excel$_{TM}$ spreadsheet is provided in Appendix H that accepts a desired moment set and provides Coxian parameters when $c > 1$.

F.10 Conclusions

To summarize results in this section, Johnson and Taaffe showed that it is possible to match three moments with a mixture of two Erlang-r distributions if r is large enough to make the inequalities in Equations (66) true. Further, they provided a method to obtain the coefficients of that distribution. This leads by Cumani's result to analytical expressions for the parameters of a Coxian-$2r$. Johnson and Taaffe further proved that $r+1$ Coxian phases were sufficient to match the first three moments, but analytical expressions for the parameters have not been obtained, nor has the use of $r+1$ phases been proved to be necessary. The parameters required can be obtained easily using an NLP, however.

The bounds on the 3-moment space of a Coxian-2 were established for $c < 1$. The Coxian-2 may be generalized to the Cox-plus-Erlang-r distribution, and the moment space of each member of the family has similar properties, the most salient of which is that the feasible region when $c < 1$ is too small to be of use in moment-matching applications.

However, when $c > 1$, the Cox-plus-Erlang-r distribution is highly useful for moment-matching. Tight bounds for the distribution were established, and these are precisely the conjectured bounds for a Coxian-$(r+1)$ distribution when $c > 1$. Thus, the family of Cox-plus-Erlang-r distributions cover the feasible 3-moment

space for $c > 1$, providing a fast and easy approach to 3-moment matching for Coxian distributions.

If $c \leq 1$, three moments are often not required [3]. If they are, a reasonable approach to obtaining Coxian parameters is to match a mixture of Erlang-r and exponential distributions using Johnson's and Taafe's empirical approach [77], then to transform the result into a Coxian-$(r+1)$ distribution by means of Cumani's result [29].

Three other tools were developed. A new coordinate system was proposed to examine the moment space, one in which bounds for Coxian distributions are straight lines. A recursive expression was devised to calculate moments of Coxian distributions efficiently. A graphical approach to transforming phase distributions via Cumani's result was employed. Each of these tools were of use in this research and may be of use to others.

Appendix G. Calculating the Exponential of a Matrix

Section 3.3 developed an algorithm for the evaluation of the cost of an appointment using phase-type approximations of service distributions. This algorithm depends on the evaluation of $\exp[Q(\tau_{j+1}-\tau_j)]$, where Q is an $(N+1)\times(N+1)$ matrix. While a number of approaches to this calculation have been advanced in the past, some have poor accuracy or are inefficient for matrices with particular features [108]. The goal of this appendix is to ascertain a method of evaluation appropriate for the cost evaluation algorithm. In passing, several problems encountered in commercial software packages will be examined as well.

Several features of the problem at hand are pertinent.

- As discussed in Section 3.3 it is desired to calculate an exponential only once for each appointment system, since this is the most computationally intense part of the cost algorithm. The exponential must be determined for each arrival, but if each $\tau_{j+1}-\tau_j$ is a multiple of some Δ, then $\exp(Q(\tau_{j+1}-\tau_j))$ is an integral power of $\exp(Q\Delta)$, making it simpler to find each exponential once $\exp(Q\Delta)$ is determined. When arrival times are lattice, Δ, the smallest step size, is given. In the continuous arrival time case, the algorithm in Section 4.4 discretizes the problem with increasingly small values of Δ, and the same approach holds. For problems with nonlattice arrival times, a lattice approximation could be imposed, making Δ the greatest (nearly) common integer divisor of $\tau_2 - \tau_1$, $\tau_3-\tau_1, \ldots \tau_{n+1}-\tau_1$. Thus, a desirable feature of the calculation of $\exp(Qk\Delta)$ is that the approach allow recalculation with different k with minimal additional computation. At worst, one may calculate $\exp(Q\Delta)$, then take the resulting matrix to the k^{th} power; some methods (*e.g.*, Cayley-Hamilton) may involve less computation than this.
- Since Q is a probability matrix, one may first define P as the transition matrix obtained by eliminating the exit state from the system, then calculate the $N\times N$

matrix $\exp(P\Delta)$. The matrix $\exp(Q\Delta)$ can then be formed by appending a row of N zeros to the bottom of $\exp(P\Delta)$, then calculating the $(N+1)^{st}$ column so that the sum of all rows is unity. This may improve the efficiency of some methods slightly, since they are operating on a smaller matrix.

- Q is triangular. For triangular matrices, the eigenvalues, required for some exponentiation methods, are equal to the diagonal elements. This immediate access to the precise eigenvalues may lend some advantage to methods that require them. The further observation that P often is sparse over entries far from the diagonal does not seem to lend any computational advantages here.

- The function $\exp(Q)$ is well-defined. If a function of a complex scalar has a Maclaurin expansion that converges for all arguments in an open ball of some radius about the origin, the series also converges for any argument that is a square matrix and for which the largest eigenvalue lies in the same ball [16]. Such a function is called "well-defined" within that ball. It can be shown that, since Q is upper triangular, any well-defined function of Q is also upper triangular, simplifying calculations [124].

- The application of some of the moment-matching approaches in Appendix F can result in phase rates that are widely disparate. This is particularly true of the Coxian-$(r+1)$ distribution formed by appending a Coxian phase to an Erlang-r. Since these phase rates are the negatives of the eigenvalues (with multiplicity), an exponentiation approach is required that can handle widely varying eigenvalues.

- Due either to use of moment-matching approaches or to a number of identical customers, it may be expected that a number of phases are identical. Many exponentiation methods encounter numerical instabilities when eigenvalues are confluent (identical) or nearly so.

- The problem size of interest will be limited arbitrarily to fewer than 30 customers, each represented by a Coxian distribution with 3 or fewer phases, so the algorithm should be capable of exponentiating matrices with $N < 100$.

With these features and caveats in mind, a number of approaches were considered . Three were coded in FORTRAN-90, and some use IMSL$_{TM}$ routines. The listings appear in Appendix H. Three routines provided in the MATLAB$_{TM}$ software package were considered as well. Several other methods were examined theoretically, in light of the success or failure of the application of those tested.

G.1 Maclaurin Series Truncation

The Maclaurin series approach is based on truncation of

$$\exp(P\Delta) = \sum_{j=0}^{\infty} (P\Delta)^j / j! \tag{73}$$

which is the definition of the exponential of a matrix. $(P\Delta)^0$ is defined to be the identity matrix of appropriate size (here, $N \times N$). This series converges for all arguments.

Approaches to matrix exponentiation based on truncation of the Maclaurin series have a long history. Error analyses, critical for effective truncation, were provided by Standish [151] and by Liou [100]. Let $T_k(P\Delta)$ be the series of the first k terms. Liou showed that

$$\|\exp(P\Delta) - T_k(P\Delta)\|_2 \leq \frac{\|P\Delta\|_2^{k+1}}{(k+1)!\left[1 - \|P\Delta\|_2/(k+2)\right]} \tag{74}$$

Here, $\|A\|_2$ is the 2-norm, which is defined as follows: [54]

$$\|x\| = \left[\sum_{j=1}^{N} |x_j|^2\right]^{\frac{1}{2}}$$

$$\|A\|_2 = \sup_{\|x\|\neq 0}\left[\frac{\|Ax\|}{\|x\|}\right] = \sup_{\|x\|=1}\|Ax\| \tag{75}$$

where x is an N-vector. This 2-norm is calculation-intensive, and it may be worth the loss in bounding effectiveness to replace it with the Frobenius norm when N is small [17]:

$$\|A\|_2 \leq \|A\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}|A_{ij}|^2} \leq \sqrt{N}\|A\|_2 \tag{76}$$

There apparently has been no discussion in the literature about problems implementing this error bound. Specifically, when k approaches $\|P\Delta\| - 2$, the denominator of Equation (74) approaches zero, causing the algorithm to stop, no matter how large the error term. On the other hand, as long as $\|P\Delta\|$ is not very close to any integer greater than one, this situation never occurs. An easy way to avoid it is to require $\|P\Delta\| \ll 2$. More will be said on the issue of the magnitude of $\|P\Delta\|$ shortly.

Standish showed (in a larger context) that, if P is a transition matrix,

$$\|\exp(P\Delta) - T_k(P\Delta)\|_T \leq \left(\frac{2^{k-1}\left(\|P\Delta\|_T\right)^k}{k!}\right) \tag{77}$$

where $\|P\Delta\|_T$ is the Tchebycheff norm of $P\Delta$ in $\Re^{N^2}$, defined by

$$\|P\Delta\|_T = \max\left(|[P\Delta]_{ij}|\right) \quad \text{over} \quad i,j = 1,2\ldots N \tag{78}$$

Golub and Van Loan point out that $\|P\Delta\|_T \leq \|P\Delta\|_2 \leq N\|P\Delta\|_T$, leading to the possibility of roughly comparing the error bounds of Standish and Liou. Suppose $\|P\Delta\|_T = \|P\Delta\|_2$, and assume for now that $\|P\Delta\|_2 < 1$. If k is fixed, two monotonically increasing error bounds result. An example is seen in Figure 38 for the case of $k = 2$. Liou's bound on the error is smaller (and hence more accurate) for all but very small values of $\|P\Delta\|_2$. For larger values of k, the crossover point where both bounds are equal increases, but does not exceed $\|P\Delta\|_2 = 0.2$ when $k < 100$.

Figure 39 shows a log-log plot of the error bounds for $\|P\Delta\|_2 = 0.1$. While it shows Standish's bound dominating for much of the range of k, the amount of dominance is masked by the log scale. Even for those values of $\|P\Delta\|_2$ and k for which Standish's bound is superior, it is not substantially so, if $\|P\Delta\|_T = \|P\Delta\|_2$.

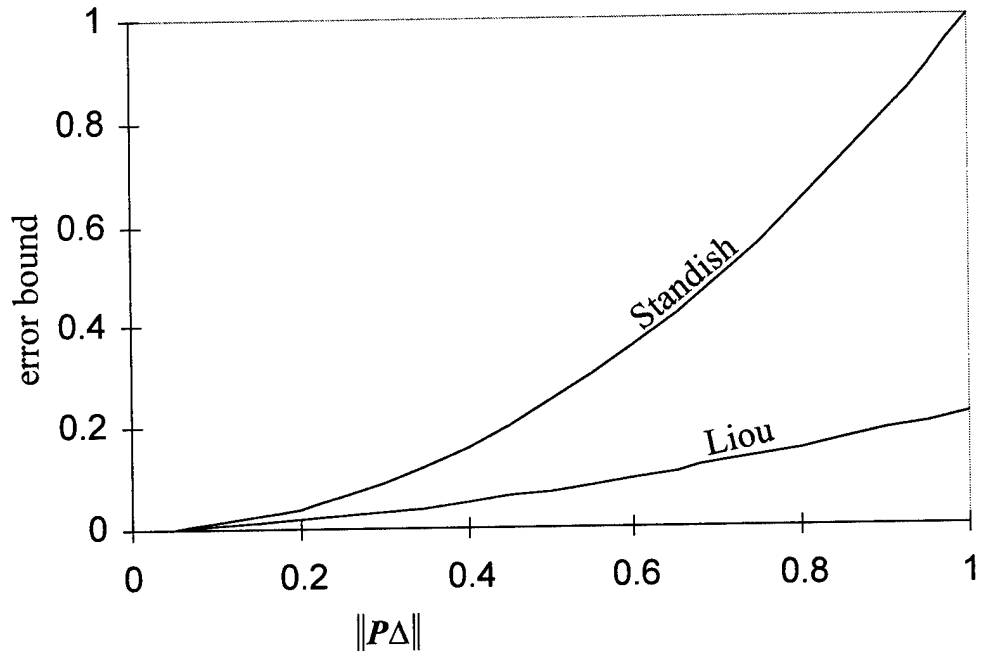


Figure 38. Error bounds versus the norm for a Maclaurin series approximation when $\|P\Delta\|_T = \|P\Delta\|_2 = 0.1$.

Requiring $\|P\Delta\|_T = \|P\Delta\|_2$ puts the ratio of Standish's bound to Liou's bound at its highest value over all possible $N \times N$ matrices. At its minimum possible value, the ratio is a factor of N smaller. For sufficiently large N and the right P, then, Standish's bound could be substantially superior to Liou's, regardless of the size of k and $\|P\Delta\|_T$. The analysis shows that neither bound dominates in all cases, so a stopping criterion for the Maclaurin exponentiation program in Section H.1 is formed from the combination of the two.

Moler and Van Loan pointed out that algorithms based on series truncation are typically ineffective for large values of $\|P\Delta\|$ [108], true regardless of how the norm is computed. If $\|P\Delta\| > 1$, there is a "hump" in the graph of $\|\exp(P\Delta)\|$

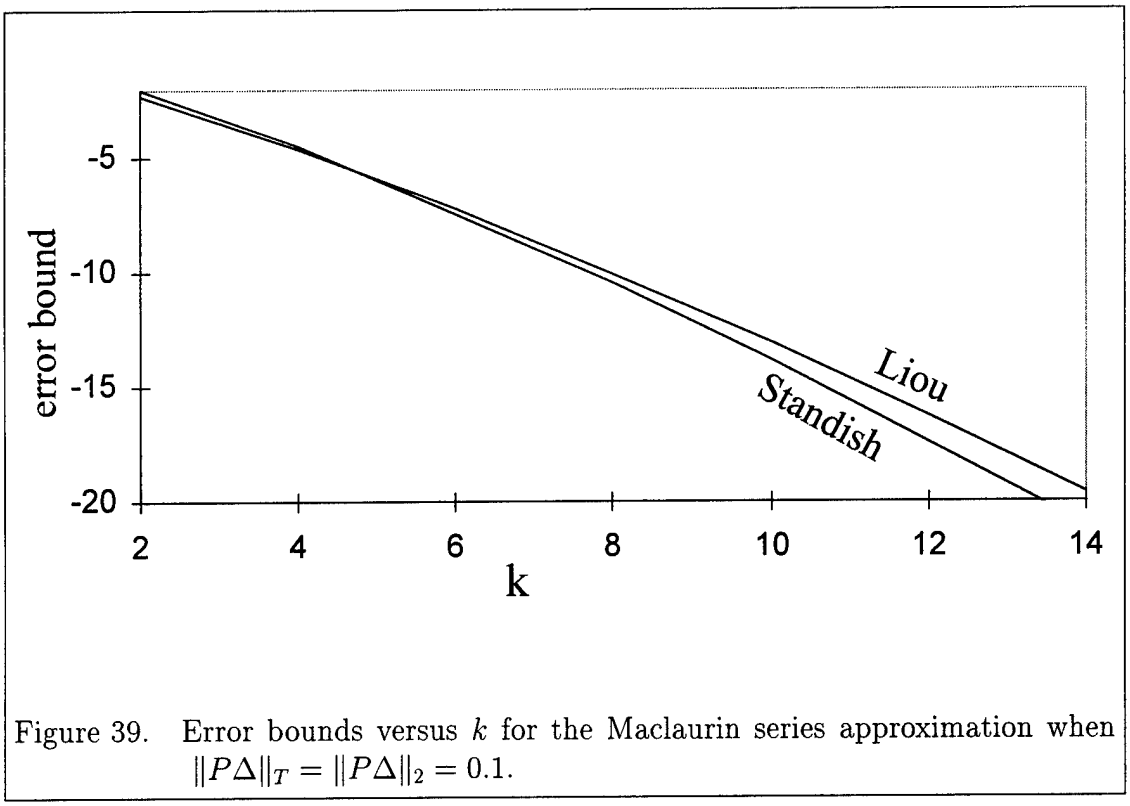


Figure 39. Error bounds versus k for the Maclaurin series approximation when $\|P\Delta\|_T = \|P\Delta\|_2 = 0.1$.

vs $\|P\Delta\|$ that increases the number of terms needed for a given accuracy. They recommended adding a scale-and-square algorithm to such approaches. This is based on the identity $\exp(A) = (\exp(A/y))^y$. Choose m such that $\|P\Delta\ /2^m\| \leq 1$, then find $\exp(P\Delta\ /2^{2^m})$ by some method. Finally, recursively square the result m times to obtain $\exp(P\Delta)$. While formal error analysis of this scale-and-square routine has not been accomplished, it appears that little precision is lost from scaling even by 2^{16}. Moler and Van Loan reported that this scaling and squaring, combined with Maclaurin series truncation, is one of the most effective methods of matrix exponentiation known [108].

G.2 Padé Approach

This approach is based on an extension of Maclaurin series representation from polynomials expressions to ratios of polynomial expressions. The (p, q) Padé approximation to $f(x)$ is the ratio $r(x) = N_{pq}/D_{pq}$ for which each of the derivatives

of $r(x)$ is equal to the corresponding derivative of $f(x)$, where p and q are the degrees of N_{pq} and D_{pq}, respectively. For exponentiation, it can be shown that [98]

$$N_{pq}(Q) = \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} Q^j$$
$$D_{pq}(Q) = \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} Q^j \tag{79}$$

The most common Padé approach to exponentiation is to set $p = q$ (diagonal approximation), first applying a scale-and-square procedure similar to that above, and for the same reasons. The restriction to diagonal approximations is particularly important when Q has widely divergent eigenvalues [108]. An algorithm is provided in Golub and Van Loan [54], and a corresponding program is provided in the MATLAB$_{TM}$ software package. This program is employed in the analysis below.

G.3 Cayley-Hamilton Approach

A corollary of the Cayley-Hamilton theorem states that every well-defined function of an $N \times N$ matrix can be expressed precisely as a polynomial function of the matrix of degree $N-1$:

$$\exp(P\Delta) = \sum_{j=0}^{N-1} a_j (P\Delta)^j \tag{80}$$

The a_j are complex scalars and are defined by the system of equations

$$\exp(\lambda_i) = \sum_{j=0}^{N-1} a_j (\lambda_i)^j \tag{81}$$

Here, the λ_i are the distinct eigenvalues of $P\Delta$. If λ_i has multiplicity m, the additional equations

$$\exp(\lambda_i) = \frac{\partial^k}{\partial x^k} \sum_{j=0}^{N-1} a_j (x)^j \Bigg|_{x=\lambda_i} \qquad j = 1, 2, \ldots m \tag{82}$$

also hold. These represent the j^{th} derivatives of each side of Equation (81). The equations are inverted to obtain each a_j, and the exponential can then be found using Equation (80).

Such an approach appears highly suitable for numerical determination of the exponential when N is small; there is no truncation error, and the most complex numerical operations are inversion of an $N \times N$ matrix and finding powers of $P\Delta$. It would seem that, since the only source of error is in these simple calculations, precision would be very good.

Unfortunately, when calculating Equation (80) on a floating-point machine, there is a source of error sometimes called "catastrophic cancellation" [108]. Suppose that one is working with word size of 15 decimal digits (typical for double-precision machines) and that two terms of this equation are very nearly additive inverses. If the series should sum to a number that is a factor of 10^j smaller in magnitude than the magnitude of its maximum term, one may not reasonably expect precision of more than $15 - j$ digits. This catastrophic cancellation is inherent in the Cayley-Hamilton approach and becomes a common problem when the eigenvalues span several powers of ten.

As a result, the precision of the method does not reach that of the floating point word size. Precision is defined here as the maximum error over all elements of the exponential. In the exponentiation program in Section H.3, the precision of the Cayley-Hamilton approach is measured roughly by finding the maximum ratio obtained by dividing each of the series terms of a matrix element by the element, then taking the maximum of this quantity over all the elements. The number of digits of precision is taken to be the difference of the number of digits in the word size and the base 10 logarithm of the maximum ratio. The user is alerted when this procedure detects catastrophic cancellation. This approach is heuristic; it will not catch all severe errors, but will catch many.

Precision is also lost when eigenvalues are confluent or nearly so, apparently due to the inversion of the matrix of coefficients. Examples of both these types of errors will be shown below.

G.4 Jordan Approach

A number of decomposition approaches are possible, based on the fact that if $P\Delta = SBS^{-1}$, then $\exp(P\Delta) = S\exp(B)S^{-1}$ [108]. The methods strive for two conflicting objectives: forcing B to be close to diagonal so that $\exp(B)$ is easy to calculate, and making S well-conditioned so that errors are not magnified [108]. The two most commonly encountered approaches are the Jordan canonical form and the Schur transformation. The Jordan transformation emphasizes the first objective, while the Schur transformation emphasizes the second.

In the Jordan transformation, the goal is to obtain B in Jordan form, in which the matrix is block diagonal, with each block being either diagonal or bidiagonal, with its diagonal elements equal and its superdiagonal elements all equal to one. In this form, $\exp(B)$ is easy to calculate analytically, and the difficult part is in finding S. Wang depended heavily on this approach to matrix exponentiation for evaluation of appointment schedule costs as a means of simplifying calculations [163]. A version of this approach is provided in the MATLAB software, and this was used for comparison purposes.

The approach is capable in theory of dealing with precisely confluent eigenvalues. However, Parlett pointed out that if $P\Delta$ is defective, B is a discontinuous function of the eigenvalues of $P\Delta$ [123]. Thus, in situations in which eigenvalues are confluent or nearly so, a small roundoff error results in a very large change to $\exp(P\Delta)$. As will be seen, unrealistic results are frequently generated by this method.

G.5 Parlett Approach

One of the convenient forms sought when employing decomposition methods is upper triangular, typically obtained by a Schur transformation. Here, a transformation is unnecessary, since $P\Delta$ is already of this form. Once in this form, the matrix is commonly exponentiated using Parlett's approach. Parlett demonstrated a simple recursive relation for well-defined functions of block-triangular matrices, of which triangular matrices are a special case. Define $T = P\Delta$ and $F = \exp(T)$. The relation is based on the facts that F has the same block structure as T and that $FT = TF$. From these two properties, it can be shown that [124]

$$\begin{aligned} F_{r,r} &= \exp(T_{r,r}) \quad \text{for } r = 1, 2, \ldots N \\ T_{r,r}F_{r,s} - F_{r,s}T_{s,s} &= \sum_{k=0}^{s-r-1} (F_{r,r+k}T_{r+k,s} - T_{r,s-k}T_{s-k,s}) \quad \text{for } r < s \end{aligned} \tag{83}$$

where $F_{r,s}$ may be an element or a rectangular block. Like the Cayley-Hamilton approach, this approach suffers from numerical difficulties. The flaw in this particular approach is that as the difference of the i^{th} and j^{th} eigenvalues vanishes, the numerator and denominator of the expression for the (i, j) element of the exponential also both vanish (for upper triangular matrices). Repeated divisions of this type can lead to gross inaccuracies.

One approach to dealing with precisely confluent eigenvalues is to modify them slightly, still staying within the acceptable error for the problem, but ensuring they are far enough apart to avoid large errors. This approach is built into the Parlett exponentiation code in Appendix H.3, but the analysis in the next section will show that it is a dangerous way to proceed, even when using quadruple precision arithmetic.

Parlett suggested the possibility of applying a similarity transform that would reposition the confluent eigenvalues into the same upper triangular block [123]. The block could be exponentiated analytically, then the block version of Parlett's method

could be applied. This might be applicable to the problem at hand, but it would destroy the convenient upper triangular structure and create blocks that could be nearly the size of the entire matrix, putting the problem solver in a potentially worse position than when he/she started. Further, such a block scheme would only be helpful if the eigenvalues were precisely confluent. Parlett did not comment on inaccuracies arising from nearly confluent eigenvalues, which will be seen shortly to be substantial.

G.6 Selection of the Most Effective Approach

Given the goals set forth above and the problems that are inherent in the approaches discussed, the most accurate approach must be determined, with short computation time being a secondary goal. To do so, a set of benchmarks is required for which the exponential is known to sufficient accuracy. The test matrix chosen to compare results is

$$Q = \begin{bmatrix} -1.0 & 0.1 & 0.9 & 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0-\delta & 1.0+\delta & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -1.0+\delta & 1.0-\delta & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -1.0-2\delta & 0.5+\delta & 0.5+\delta \\ 0.0 & 0.0 & 0.0 & 0.0 & \lambda & -\lambda \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{84}$$

With appropriate choice of parameters δ and λ, this matrix provides some insight into the behavior of the various methods in the presence of confluent or nearly confluent eigenvalues, as well as eigenvalues that are widely separated.

For the Maclaurin, Cayley-Hamilton, and Parlett approaches, the programs in Appendix H were employed on a Pentium$_{TM}$ processor using double precision arithmetic, which at best can give 14 or 15 places of precision. For the Padé and Jordan

Table 23. Accuracy of $\exp(Q)$ when some eigenvalues are nearly confluent

δ	Parlett	Cayley-Hamilton	Jordan	Maclaurin	Padé
10^{-2}	13	6	12	14	14
10^{-3}	9	6	9	14	14
10^{-4}	6	5	4	14	13
10^{-5}	2	2	3	14	14
10^{-6}	0	0	0	14	13
10^{-7}	0	0	0	14	14
10^{-8}	0	0	0	14	12

approaches, routines supplied with the MATLAB software package were employed. These also used the Pentium's double precision arithmetic routines.

First, the question of nearly confluent eigenvalues was addressed by fixing $\lambda = -20$ and modifying δ. Since analytic exponentiation of this matrix is oppressive, results using the Maclaurin approach with quadruple precision were employed as the benchmark. Results are in Table 23. Entries refer to the maximum number of decimal places for which every matrix entry agrees with the actual result, without rounding.

Consider the results of Parlett's method. For $\delta = 10^{-5}$, if $E = \exp(Q)$, then $E(1,4)$ should be approximately 0.17167. To obtain $E(1,4)$, Parlett's method performs four exponentiations initially to obtain the diagonal elements of E, after which it performs only 21 additions and subtractions, 14 multiplications, and 6 divisions. Using double precision arithmetic yields 0.172213, in error by 0.3%. Clearly impossible results ensue for smaller δ. The same calculation in quadruple precision yields only 5 places of accuracy when $\delta = 10^{-10}$ and fails completely when $\delta = 10^{-12}$.

Moler and Van Loan commented that it would be interesting to compare the Parlett approach to a scaling-and-squaring approach, and they hinted that the Parlett approach would prove more accurate [108]. The above results for the Maclaurin and Padé approaches indicate that the opposite is true in the presence of nearly

confluent eigenvalues. Because of this degradation, Parlett's approach will not be considered further.

Likewise, The Cayley-Hamilton method yields $E(1,4) = 0.171156$, in error by 0.3%, showing its susceptibility to nearly confluent eigenvalues as well. The Jordan method was equally poor. Conclusions regarding the relative effectiveness of these two algorithms should be avoided, since they were implemented in different languages by different people. However, it is clear that both approaches are inadequate for the purpose at hand. This poor performance of the Jordan algorithm in the presence of nearly confluent eigenvalues was anticipated by Moler and Van Loan [108].

The Maclaurin and Padé approaches used nine and six terms, respectively, to obtain each of the results in Table 23. More terms did not appear to help achieve better convergence, although as few as three terms were required to obtain 14-place accuracy in some cases. These methods appear highly accurate and insensitive to the presence of nearly confluent eigenvalues.

The analysis above used the results of a quadruple precision Maclaurin routine as a benchmark. This is inelegant and may lead to problems if the routine itself is suspect. For the majority of matrices, this may be the only alternative, since analytic results are difficult to obtain and result in very long expressions.

If some regularity is present in Q, however, analytical results may be obtainable, albeit with some effort. Consider the matrix formed when $\delta = 0$:

$$Q = \begin{bmatrix} -1.0 & 0.1 & 0.9 & 0.0 & 0.0 & 0.0 \\ 0.0 & -1.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -1.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -1.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.0 & \lambda & -\lambda \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{85}$$

Exponentiation can be performed analytically most simply in this case by first making the following definitions:

$$Q = \left[\begin{array}{c|c} a & b \\ \hline 0 & c \end{array}\right] \qquad E = \exp(Q) = \left[\begin{array}{c|c} A & B \\ \hline 0 & C \end{array}\right] \tag{86}$$

where a is 4×4. Parlett's result can be used to obtain

$$A = \exp(a) \quad C = \exp(c) \tag{87}$$

$$Ab - bC \;\; = aB - Bc \tag{88}$$

The first two equations are solvable analytically by the Cayley-Hamilton approach, after which the third produces a series of eight simultaneous equations that can be solved analytically by sequential substitution. The results are

$$A = \frac{1}{\mathrm{e}} \begin{bmatrix} 1 & \frac{1}{10} & \frac{19}{20} & \frac{7}{15} \\ 0 & 1 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad C = \begin{bmatrix} \mathrm{e}^{\lambda} & 1 - \mathrm{e}^{\lambda} \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{-1+\mathrm{e}^{\lambda+1}}{2e(1+\lambda)} & \frac{-(2\lambda+1)+2e(1+\lambda)-\mathrm{e}^{\lambda+1}}{2e(1+\lambda)} \\ \frac{-(2+\lambda)+\mathrm{e}^{\lambda+1}}{2e(1+\lambda)^2} & \frac{-(4\lambda^2+7\lambda+2)+2(\lambda+1)^2-\mathrm{e}^{\lambda+1}}{2e(1+\lambda)^2} \\ \frac{-(\lambda^2+4\lambda+5)+2\mathrm{e}^{\lambda+1}}{4e(1+\lambda)^3} & \frac{-(10\lambda^3+29\lambda^2+26\lambda+5)+4e(1+\lambda)^3-2\mathrm{e}^{\lambda+1}}{4e(1+\lambda)^3} \\ \frac{\left(\begin{array}{c} -(28\lambda^3+141\lambda^2+258\lambda+151) \\ +6\mathrm{e}^{\lambda+1}(10+9\lambda) \end{array}\right)}{120e(1+\lambda)^4} & \frac{\left(\begin{array}{c} -(302\lambda^4+1180\lambda^3+1671\lambda^2+950\lambda+151) \\ -6\mathrm{e}^{\lambda+1}(10+9\lambda)+120e(1+\lambda)^4 \end{array}\right)}{120e(1+\lambda)^4} \end{bmatrix} \tag{89}$$

As long as $|1 + \lambda|$ is not very small, the accuracy of these expressions should be near the word size on most floating-point machines. These expressions were used to test the candidate approaches for robustness under widely differing eigenvalues, as well

Table 24. Accuracy of $\exp(Q)$ as an eigenvalue diverges. Number of decimal places of accuracy are given.

λ	Maclaurin(10)	Maclaurin(9)	Maclaurin(4)	Padé(6)	Padé(3)
$-2 \cdot 10^{-8}$	12	10	3	14	9
$-2 \cdot 10^{-1}$	12	10	3	14	9
$-2 \cdot 10^{0}$	12	10	3	14	14
$-2 \cdot 10^{1}$	14	13	7	13	13
$-2 \cdot 10^{2}$	14	12	10	12	13
$-2 \cdot 10^{3}$	13	11	13	12	13
$-2 \cdot 10^{4}$	12	11	12	11	11
$-2 \cdot 10^{5}$	11	11	11	11	10
$-2 \cdot 10^{6}$	11	11	11	9	9
$-2 \cdot 10^{7}$	11	11	11	8	8
$-2 \cdot 10^{8}$	9	8	8	8	8

as in the presence of precisely confluent eigenvalues. The results are shown in Table 24.

While not included in Table 24, for $\lambda \in [-2000, -1]$, the Cayley Hamilton algorithm achieved relatively constant maximum error of $6 \cdot 10^{-7}$. For $\lambda < -2 \cdot 10^{-5}$, the routine gave unrealistic results.

For an accuracy of 10^{-7}, Liou's rule requires ten terms of the Maclaurin series for each value of λ, which is far more than necessary for most values. The Maclaurin truncation rules were obviated for this analysis, and the MATLAB Padé routine was modified, to enable one to compare results using a constant number of terms. The number of terms used in each calculation is shown in parentheses in each column of Table 24.

The evaluations of a small number of terms of the two series reveals that convergence is very quick over a range of λ, but is slower above and below this value. Accuracy becomes more uniform over a broader range of λ as the number of terms increases.

MATLAB employs the Padé algorithm offered by Golub and Van Loan, who point out that it requires on the order of $2(k + m + \frac{1}{3})N^3$ flops. A flop is defined as

a single floating-point operation, while k is the number of terms evaluated and m is the number of half-scalings required [54]. This is for a general matrix; since the matrices here are triangular, only $\frac{1}{3}N^3$ rather than $2N^3$ flops are needed for each matrix multiplication, reducing the Padé requirement to $\frac{1}{3}(k+m+1)N^3$ [54]. The Maclaurin series approach also requires on the order of $\frac{1}{3}(k+m+1)N^3$, as it turns out. Thus, one can compare the algorithmic efficiency of the two by considering the number of terms required by each to achieve a given accuracy.

For the family of matrices analyzed here, the number of terms of the Maclaurin series to reach the accuracy of the 6-term Padé series was 9, with between 0 and 6 scalings required. Thus, the relative efficiency of the Padé and Maclaurin approaches varied between $\frac{(9+0+1)}{(6+0+1)} \approx 1.43$ and $\frac{(9+6+1)}{(6+6+1)} \approx 1.23$. Moler and Van Loan reported that, when combined with a scale-and-square routine, the Padé approach achieved similar precision to that of the Maclaurin series approach using approximately half as many terms [108]. This would imply a higher relative efficiency for the Padé approach than that calculated here, but still less than 2.0.

The Padé approach has been shown here to be more efficient for representative matrices, possibly 1.2 to 2 times so. However, while the improvement in efficiency may make it worth coding the Padé approach for some problems, it appears that the Maclaurin series approach is adequate for the matrices to be dealt with here, and it will be employed.

Other approaches are possible. Since the matrix represents a system of differential equations, Runge-Kutta approximation methods can be used. Lagrange interpolation is also possible, which yields expressions similar to those obtained by Parlett's method. Engineers often use a method based on inverse Laplace transforms, which turns out to be related to the Cayley-Hamilton approach [108]. Each of these approaches has been found deficient by other researchers, so they were not tested [108].

G.7 Software Concerns

Several observations may be of use to the analyst using mathematical software for the PC (personal computer). Floating point problems were observed in Microsoft's PowerStation$_{TM}$, a FORTRAN compiler for PCs. Errors as high as 40% originally were obtained for the test matrix using the Parlett program. These stemmed from two sources. One is the documented problem with all versions of this compiler in which double precision numbers must be initialized with double precision constants; the statement $a = 0.$ causes errors in the last 32 of the 64 bits in a double-precision word, while $a = 0.0\mathrm{D} + 0$ does not. The error arises whether using a 32-bit or 16-bit processor. However, the majority of the error was caused by an undocumented error generated when double and single precision numbers are used in the same calculation. The manufacturer has acknowledged this error [106]. After modifying the code to eliminate these compiler errors, results were identical to those obtained on other compilers [23].

The MATLAB software package has two problems in its matrix exponentiation routines. As noted above, its implementation of Jordan decomposition gives unacceptably high errors in tests on the test matrix. Worse, one can obtain negative and even infinite results without invoking a warning from the error-handling routine in the program. The manufacturer considers this program to be of pedagogic value only [109].

In tests of MATLAB's Maclaurin series algorithm on the test matrix, the performance was poor compared to the FORTRAN subroutine EXP in Section H.1. For $\lambda \in [-200, -1]$, its precision was only 10^{-6}. For more negative values of λ, unrealistic results were obtained. Addition of a scale-and-square routine to the program produced precision near 10^{-12} for all values of λ tried. The efficacy of this simple change has been acknowledged by the manufacturer [109].

Due to previously reported problems with versions of the Pentium processor and the need to track down numerical anomalies (which transpired to be the com-

piler errors discussed above), floating point results from the Pentium and a Sun SparcStation$_{TM}$ workstation were compared. Both use two 32-bit words for double precision storage and manipulation. The author conjectured that, since the Pentium has several rounding modes and the default for FORTRAN compilers is its truncation mode, perhaps the last digit might differ. This could create substantial differences in final results, since routines like the Parlett algorithm are only marginally stable, as shown above. Even after a number of operations designed to degrade accuracy to only five places, the double precision arithmetic results produced by the Sun FORTRAN compiler and those produced internal to the Pentium were the same (albeit incorrect) to 15 digits. It appears that the Pentium does indeed conform to IEE Standards 754 and 854 for floating point storage and arithmetic [67].

Appendix H. Computer Programs

H.1 Sequence/Schedule Optimization Program

FORTRAN code: Following are the programs referenced in this dissertation. These particular versions of the routines are the ones used in the validation of the greedy sequencing algorithm and are presented in top-down order. For ease in reference, Table 25 shows the structure of the program, with brief descriptions of the function of each major routine and the page number on which it can be found.

Table 25. Program structure and subroutine index

- COMMONS (Page 201): Used by INCLUDE statements throughout.
- HCSEARCH (Page 202): Program to compare the global optimum and the optimum found by the greedy sequencing algorithm for a series of problems.
  - FPFLAW (Page 226): Checks for Pentium chip flaw.
  - NORMDIST (Page 208): Returns a normally distributed variate.
  - RECORD1 (Page 207): Records data used in each run.
  - SEQUENCE2 (Page 209): Shell that takes the place of a sequencing program used in another version of the code.
  - FIXEDLATTICE (Page 210): Finds the optimal schedule for a given sequence.
    - BUILDQ (Page 221): Builds the transition matrix Q.
      - MATMULT (Page 225): Multiplies two upper-triangular matrices.
    - OMEGAEVAL (Page 222): Builds the conditional probability matrix Ω.
    - EXP (Page 223): Performs matrix exponentiation.
    - EVALUATE (Page 214): Performs cost evaluation algorithm.
      - PEXTEND (Page 216): Modifies waiting times using Equation (7).
    - FATHOM (Page 211): Finds S_E (S_L) using the fixed-lattice algorithm.
    - ENUMERATE (Page 212): Evaluates the necessary schedules between S_E and S_L to determine the optimum.
      - FLIP (Page 214): Recursive subroutine performs binary enumeration.
      - EVALUATE (Page 214): Performs cost evaluation algorithm.
  - GREEDYSEQ (Page 217): Performs the greedy sequencing algorithm.
    - BUILDSAME (Page 219): Generates matrix indicating if two customers are of the same class. IF so, GREEDYSEQ need not swap them.

MODULE SETSIZE

```
!variables shared by most routines to set array sizes
!MAXCUST and MAXPHASES are maximum number of customers and phases
!allowable in problem.  They need be changed only here.  Execution
!speed is dep on these variables, so set them small as practicable.
INTEGER LDA,MAXCUST,MAXPHASES,K,N,OT
PARAMETER(MAXPHASES=4)
PARAMETER(MAXCUST=6)
PARAMETER(LDA=(MAXPHASES+1)*(MAXCUST+1))
PARAMETER(MAXSEQ=721)
REAL*8 DELTA
COMMON /SIZEDATA/ K,N,OT,DELTA
END MODULE
```

MODULE SETQDATA

```
!variables shared by cost-evaluation-level routines
USE SETSIZE
INTEGER NQ,R(MAXCUST)
REAL*8 MU(MAXCUST,MAXPHASES),B(MAXCUST,MAXPHASES)
REAL*8 GAMMA(MAXCUST+1)
REAL*8 Q(LDA,LDA),E(LDA,LDA),OMEGA(MAXCUST+1,LDA)
COMMON /QDATA/ NQ,R,GAMMA,OMEGA,Q,E
END MODULE
```

MODULE SETCOSTDATA

```
!variables shared by schedule-optimization-level routines
USE SETSIZE
INTEGER NIT,FLAG,OTSLOT
REAL*8 CW(MAXCUST+1),HORIZON,OTPOINT
COMMON /COSTDATA/ CW,NIT,FLAG,HORIZON,OTSLOT,OTPOINT
END MODULE
```

MODULE SETSEQDATA

```
!variables shared by sequence-optimization-level routines
USE SETSIZE
PARAMETER(ACCURACY=1.0D-6)
INTEGER RSAVE(MAXCUST),SBEST(MAXCUST),NBEST
INTEGER SQ(MAXSEQ,MAXCUST),NSEQ,NINIT(2)
REAL*8 MUSAVE(MAXCUST,MAXPHASES),BSAVE(MAXCUST,MAXPHASES)
REAL*8 GAMMASAVE(MAXCUST+1),CWSAVE(MAXCUST+1)
REAL*8 C(0:MAXSEQ),WBEST(MAXCUST+1)
CHARACTER*10 ALPH(MAXSEQ)
COMMON SEQDATA/ALPHINIT,NINIT,ALPH,C,SQ,MUSAVE,BSAVE,
1   GAMMASAVE,CWSAVE,RSAVE,SBEST,WBEST,N
END MODULE
```

MODULE EXPERIMENTDATA

```
!variables shared by experiment-level routines
USE SETSIZE
INTEGER NEXPT,MAXPICK,NPICK
PARAMETER(MAXPICK=7)
REAL*8 PHI2(MAXCUST),PHI3(MAXCUST),MEAN(MAXCUST)
REAL*8 PICKPHI2(MAXPICK),PICKPHI3(MAXPICK)
REAL*8 PICKR(MAXPICK),PICKMU1(MAXPICK)
REAL*8 PICKMUR(MAXPICK),PICKB1(MAXPICK)
COMMON /EXPTDATA/NEXPT,PHI2,PHI3,MEAN
END MODULE

!*******************************
```

PROGRAM HCSEARCH

```
!varies problem parameters and finds optimum by exhaustive
!enumeration.  Then it uses a greedy algorithm from two
!starting points to approximate the optimum.  Parameters altered
!in this version are CW(N+1) and HORIZON, keeping number
!of slots constant.

!inputs:
!SEQUENCE contains each possible sequence (alphanumeric)
!DISTRIB contains Coxian parameters for specific 3-moment sets

!outputs:
!OUT contains parameters used, optimum and near-optimal
!COUNTER.DAT contains results of greedy algorithm

USE MSFLIB
USE SETCOSTDATA
USE SETQDATA
USE SETSEQDATA
USE EXPERIMENTDATA

INTEGER I,J,NSOFAR,VARSEQ(MAXCUST)
INTEGER*2 IHR,IMIN,ISEC,I100
CHARACTER*10 HEADER*80, ALPHABET
REAL*8 VARMEAN,ERR,TEMP,TSTART, TSTARTINI,WVAR(MAXCUST)

CALL FPFLAW
OPEN(1,FILE='DISTRIB')
OPEN(2,FILE='SEQUENCE')
OPEN(3,FILE='OUTFULL')
```

```
      OPEN(4,FILE='OUTMEAN')
      OPEN(5,FILE='OUTRAND')
      OPEN(6,FILE='ENUMOUT')
      OPEN(7,FILE='OUTVAR')
      ALPHABET='ABCDEFGHIJ'
      NEXPT=0
      K=11

      WRITE(*,*)'STARTING EXPERIMENT NUMBER:'
      READ(*,*)NEXPT
      NEXPT=NEXPT-1

      WRITE(*,*)'NUMBER OF CUSTOMERS:'
      READ(*,*)N

      WRITE(*,*)'RANDOM SEED:'
      READ(*,*) I
      VARMEAN=RAND(I) !initialize random stream
      WRITE(3,*)'SEED= ',I

      CALL GETTIM(IHR,IMIN,ISEC,I100)
      TSTART=IHR*3600+IMIN*60+ISEC+REAL(I100)/100

      !get sequences to evaluate
      READ(2,*)HEADER
      DO I=1,MAXSEQ
         READ(2,*)(SQ(I,J),J=1,N)
         IF(SQ(I,1).EQ.0)GOTO 10
         !alph is alpha equivalent of SQ, assuming services idd
         DO J=1,N
            ALPH(I)(J:J)=ALPHABET(SQ(I,J):SQ(I,J))
         END DO
      END DO
      WRITE(*,*)'WARNING: NOT ALL SEQUENCES WERE READ IN'
10    NSEQ=I-1

      !get service distribution parameters
      WRITE(3,*)
      WRITE(3,*)'HEADER ON DIST FILE:'
      READ(1,*)HEADER
      WRITE(3,*)HEADER
      READ(1,*)HEADER
      WRITE(3,*)HEADER
      WRITE(3,*)
      DO I=1,30
```

```
         READ(1,*)PICKPHI2(I), PICKPHI3(I),PICKR(I),PICKMU1(I),
     1      PICKMUR(I),PICKB1(I)
         IF(PICKPHI2(I).EQ.0.0) GOTO 20
       END DO
20     NPICK=I-1

       VARMEAN=10.0 !variance of mean distribution

       WRITE(3,*)'K= ',K
       WRITE(3,*)'VARMEAN= ',VARMEAN

       !new experiment point
30     NEXPT=NEXPT+1
       CALL GETTIM(IHR,IMIN,ISEC,I100)
       TSTARTINI=IHR*3600+IMIN*60+ISEC+REAL(I100)/100-TSTART
       TSTART=IHR*3600+IMIN*60+ISEC+REAL(I100)/100
       WRITE(*,*)'EXPERIMENT: ',NEXPT, ' TIME: ',TSTARTINI
       SUMMEANS=0

       !pick means from lognormal with log(mean) dist as N(0,VAR)
       !pick weights from lognormal with log(CW) dist as N(0,0.5)
       DO I=1,N
          CALL NORMDIST(MEAN(I))
          MEAN(I)=EXP(VARMEAN*MEAN(I))
          SUMMEANS=SUMMEANS+MEAN(I)
          CALL NORMDIST(CW(I))
          CW(I)=EXP(0.5*CW(I))
       END DO

       !sort customers by WSEPT
35     DO I=2,N
          FLAG=0
          IF(MEAN(I)/CW(I).LT.MEAN(I-1)/CW(I-1)) THEN
             TEMP=MEAN(I)
             MEAN(I)=MEAN(I-1)
             MEAN(I-1)=TEMP
             TEMP=CW(I)
             CW(I)=CW(I-1)
             CW(I-1)=TEMP
             FLAG=1
          END IF
       END DO
       IF(FLAG.EQ.1)GOTO 35

       !assign phase rates and transition probabilities.  If PHI2>1,
```

```
!then choose high PHI3 or low PHI3 with equal probability.
DO I=1,N !for each customer
   J=INT(5*RAND(0)+1)
   IF(J.GT.3) J=J+INT(2*RAND(0))*2
   PHI2(I)=PICKPHI2(J)
   WVAR(I)=(PHI2(I)-1)*MEAN(I)**2/CW(I)
   PHI3(I)=PICKPHI3(J)
   R(I)=PICKR(J)+1
   MU(I,1)=PICKMU1(J)/MEAN(I)
   B(I,1)=PICKB1(J)
   DO H=2,R(I)
      B(I,H)=1.0
      MU(I,H)= PICKMUR(J)/MEAN(I)
   END DO

   GAMMA(I)=1.0 !0.7+0.3*RAND(0)
END DO

!save settings before shuffling
CALL RECORD1(3)
RSAVE=R
MUSAVE=MU
BSAVE=B
GAMMASAVE=GAMMA
CWSAVE=CW
LASTCOST=0.0D0

DO J=-1,3
   CW(N+1)=10.0**FLOAT(J)
   DO I=1,20
      HORIZON=FLOAT(I)/10.0*SUMMEANS
      DELTA=HORIZON/FLOAT(K-1)
      OTPOINT=HORIZON
      OTSLOT=K-1
      NBEST=0
      C(0)=1.0D50

      CALL SEQUENCE2 !find cost of each sequence

      !stop incrementing horizon if the cost is tiny
      IF(C(NBEST).LT.ACCURACY) GOTO 150

      !starting sequence ordered by weighted means
      NSOFAR=1
      CALL GREEDYSEQ(ERR,ITER,NSOFAR)
```

```
      ERRFLAG=0
      IF(ERR.GT.0.0) ERRFLAG=1
      WRITE(4,140)NEXPT,C(NSOFAR),C(NBEST),ITER,HORIZON,
     1    HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)
      WRITE(3,140)NEXPT,C(NSOFAR),C(NBEST),ITER,
     1    HORIZON,HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)

      !random starting sequence
      NSOFAR=INT(NSEQ*RAND(0)+1)
      CALL GREEDYSEQ(ERR,ITER,NSOFAR)
      ERRFLAG=0
      IF(ERR.GT.0.0) ERRFLAG=1
      WRITE(5,140)NEXPT,C(NSOFAR),C(NBEST),ITER,
     1    HORIZON,HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)
      WRITE(3,140)NEXPT,C(NSOFAR),C(NBEST),ITER,
     1    HORIZON,HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)

      !order starting sequence by weighted variances
      NSOFAR=1
      DO II=1,N
         VARSEQ(II)=II
      END DO
40    FLAG=0
      DO II=2,N
         IF(WVAR(VARSEQ(II)).LT.WVAR(VARSEQ(II-1)))THEN
            FLAG=1
            TEMP=VARSEQ(II)
            VARSEQ(II)=VARSEQ(II-1)
            VARSEQ(II-1)=TEMP
         END IF
      END DO
      IF(FLAG.EQ.1)GOTO 40

      !find index of variance-ordered starting sequence
      DO NSOFAR=1,NSEQ
         DO JJ=1,N
            IF(SQ(NSOFAR,JJ).NE.VARSEQ(JJ))GOTO 50
         END DO
         GOTO 60
50    END DO
      WRITE(*,*)'VARIANCE START SEQUENCE NOT FOUND'
      WRITE(7,*)'VARIANCE START SEQUENCE NOT FOUND'
60    CONTINUE

      !perform greedy sequencing algorithm
```

```
            CALL GREEDYSEQ(ERR,ITER,NSOFAR)
            ERRFLAG=0
            IF(ERR.GT.0.0) ERRFLAG=1
            WRITE(7,140)NEXPT,C(NSOFAR),C(NBEST),ITER,HORIZON,
     1         HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)
            WRITE(3,140)NEXPT,C(NSOFAR),C(NBEST),ITER,
     1         HORIZON,HORIZON/SUMMEANS,CW(N+1),ALPH(NSOFAR)

140         FORMAT(I3,'  ',E12.5,'  ',E12.5,'  ',I3,'  ',E12.5,'  ',
     1      E12.5,'  ',E12.5,'  ', A<N>,'   ',A<N>,'  ',A23)
         END DO !I
150   END DO !J
      GOTO 30

      CLOSE(1)
      CLOSE(2)
      CLOSE(3)
      CLOSE(4)
      CLOSE(5)
      CLOSE(6)
      CLOSE(7)
      RETURN
      END

      !*********************************
```

SUBROUTINE RECORD1(FILE)

```
      !transfers input data to output file

      USE SETQDATA
      USE SETCOSTDATA
      USE SETSEQDATA
      USE EXPERIMENTDATA
      INTEGER J,I,FILE

      WRITE(FILE,*)
      WRITE(FILE,*)
      WRITE(FILE,*)
      WRITE(FILE,*)'EXPERIMENT # ',NEXPT

      WRITE(FILE,*)'CUST PHASES MEAN PHI2 PHI3 WEIGHT'
      DO J=1,N
         WRITE(FILE,50)J,R(J),MEAN(J),PHI2(J),PHI3(J),CW(J)
      END DO
```

```
50    FORMAT(I3,' ',I3,' ',E12.3,' ',E12.3,' ',E12.3,' ',E12.3)

      WRITE(FILE,*)
      WRITE(FILE,*)'PHASE RATES FOR EACH CUSTOMER:'
      DO J=1,N
         WRITE(FILE,60)J,(MU(J,I),I=1,R(J))
      END DO
60    FORMAT(I3,32E9.2)
      WRITE(FILE,*)
      WRITE(FILE,*)'TRANSITION PROBABILITIES FOR EACH CUSTOMER:'
      WRITE(FILE,*)'(FIRST LISTED IS SHOW PROBABILITY)'
      DO J=1,N
         WRITE(FILE,70)J,GAMMA(J),(B(J,I),I=1,R(J)-1)
      END DO
70    FORMAT(I3,E9.2,32E9.2)
      WRITE(FILE,*)

      RETURN
      END

      !**********************************
```

SUBROUTINE NORMDIST(X1)

```
      !returns a standard normal random variate
      !Adapted from Marsaglia and Bray,
      !SIAM Review 6:260-64, 1964

      REAL*8 X1,X2,V1,V2,Y,W
      INTEGER RESERVE
      SAVE RESERVE,X2

      IF (RESERVE.EQ.0) THEN !generate two new normal variates
         W=2.0
         DO WHILE(W.GT.1.0)
            V1=2*RAND(0)-1
            V2=2*RAND(0)-1
            W=V1**2+V2**2
         END DO

         Y=SQRT(-2*LOG(W)/W)
         X1=V1*Y
         X2=V2*Y
      ELSE !use second random variate generated from last call
         X1=X2
      END IF
```

```
      RESERVE=1-RESERVE

      RETURN
      END

      !*********************************
```

SUBROUTINE SEQUENCE2

```
      USE SETCOSTDATA
      USE SETQDATA
      USE SETSEQDATA

      INTEGER S(N+1)
      REAL*8 W(N+1)

      !  set parameters to those of sequence L
      DO L=1,NSEQ
         FLAG=0
         DO J=1,N
            R(J)=RSAVE(SQ(L,J))
            GAMMA(J)=GAMMASAVE(SQ(L,J))
            CW(J)=CWSAVE(SQ(L,J))
            DO JJ=1,MAXPHASES
               B(J,JJ)=BSAVE(SQ(L,J),JJ)
               MU(J,JJ)=MUSAVE(SQ(L,J),JJ)
            END DO
40       END DO

         !optimize schedule for customer L
         CALL FIXEDLATTICE(C(L),W,S)
         IF(C(L).LT.C(NBEST)) THEN
            NBEST=L
            SBEST=S
            !WBEST=W
         END IF

90    END DO

      RETURN
      END

      !*********************************
```

SUBROUTINE FIXEDLATTICE(C2,W2,S2)

```
      !find optimal cost and schedule for a given sequence

      USE SETCOSTDATA
      REAL*8 C1,C2,ERROR,W1(N+1),W2(N+1)
      INTEGER I,APARTFLAG,S1(N+1),S2(N+1),S(N+1),NIT1,NIT2,CHECKSUM

      !build matrix Q, determine matrix size
      !also build expected wait matrix OMEGA
      CALL BUILDQ

      !build conditional wait matrix OMEGA
      CALL OMEGAEVAL

      !find E=EXP(Q*DELTA).
      !Only need to do this when lattice size changes
      ERROR = 1.0D-7
      CALL EXP(ERROR)

      !put all customers but the first at K-1
      S1=K-1
      S=K-1
      S1(1)=0
      S(1)=0
      FLAG=0

80    CALL EVALUATE(S1,W1,C1)
      NIT=1

      !first pass
      CALL FATHOM(S1,S,W1,C1)
      NIT1=NIT

      !set second pass to start each arrival 1 slot earlier
      !than S1 if FLAG=0 and 1 slot later if FLAG=1
      DO J=2,N
         S(J)=MIN(MAX(S1(J)-1+2*FLAG,0),K-1)
         S2(J)=S(J)
      END DO

      S(1)=0
      S2(1)=0
      S2(N+1)=S2(N)
      S(N+1)=S(N)
```

```
      CALL EVALUATE(S2,W2,C2)
      FLAG=1-FLAG
      NIT=1

      CALL FATHOM(S2,S,W2,C2)
      NIT2=NIT

      !passes may not coincide, in which case
      !intermediate schedules must be enumerated
      CALL ENUMERATE(S1,S2,C1,C2,W1,W2,I,APARTFLAG,CHECKSUM)
      WRITE(6,*)NIT1,' ',NIT2,' ',NIT,' ',CHECKSUM

      RETURN
      END

      !**********************************
```

SUBROUTINE FATHOM(SOPT,S,WOPT,COPT)

```
      !Uses Simeoni-style algorithm to find early bound on minimum of
      !function "cost".  Assumes elements 1 and N+1 are fixed at
      !bounds 0 and K-1

      !FLAG SOPT is early schedule if FLAG=1, late schedule if FLAG=0
      !N length of S and SOPT, including first and last elements
      !K number of schedule slots
      !SOPT,COPT early schedule (in numbers of time slots) and cost
      !S, C current test schedule (in numbers of time slots) and cost
      !DELTA length of time step
      !M tracks which element of S is being altered
      !NIT number of cost evaluations performed

      USE SETCOSTDATA
      REAL*8 C,COPT,W(N+1),WOPT(N+1)
      INTEGER M,S(N+1),SOPT(N+1)

      !find latest customer who is not already at end of schedule
100   M=N*FLAG+(1-FLAG)*2 !M=N or M=2
      DO WHILE (S(M).EQ.(K-1)*FLAG)
         M=M+1-2*FLAG !subtract or add 1
      END DO
      IF(M.EQ.FLAG+(1-FLAG)*(N+1)) GOTO 300 !all arrivals shifted

200   S(M)=S(M)-1+2*FLAG !try to shift customer
```

```
      IF(S(M+1).GE.S(M).AND.S(M).GE.S(M-1)) GOTO 250 !if order unchanged
      IF(M.EQ.N.AND.FLAG.EQ.1) GOTO 250 !last slot can be shifted forward
      S(M)=S(M)+1-2*FLAG !no good.  undo shift
      M=M+1-2*FLAG !if not at end, try to shift next customer
      IF(FLAG*(M-2)+(1-FLAG)*(K-1-M).GT.0) GOTO 200
      GOTO 200

250   CALL EVALUATE(S,W,C)
      NIT=NIT+1
      IF(C.LT.COPT) THEN !replace SOPT with S
         SOPT=S
         COPT=C
         WOPT=W
         GO TO 100
      ELSE !undo shift -- no improvement
         S(M)=S(M)+1-2*FLAG
         M=M+1-2*FLAG !subtract or add 1
         IF(FLAG*(M-2)+(N-M)*(1-FLAG).GE.0) GOTO 200
      ENDIF

300   CONTINUE

      RETURN
      END

      !***********************************
```

SUBROUTINE ENUMERATE(S1,S2,C1,C2,W1,W2,I,APARTFLAG,
1 CHECKSUM)

```
      !enumerates all schedules between S1 and S2

      USE SETCOSTDATA
      REAL*8 C,C1,C2,W(N+1),W1(N+1),W2(N+1)
      INTEGER S(N+1),S1(N+1),S2(N+1),APARTFLAG
      INTEGER J,I,H
      INTEGER DIFFER(N),CHECKSUM
      NIT=0
      APARTFLAG=0

      !CHECKSUM is total number of positions current schedule
      !and S1 differ by
```

```
!generate list of positions in which S1 differs from S2
I=0
DO J=2,N
    IF(S1(J).NE.S2(J)) THEN
        IF(ABS(S1(J)-S2(J)).NE.1) THEN
            APARTFLAG=1
            RETURN
        END IF
        I=I+1
        DIFFER(I)=J
    END IF
END DO

!pick the lowest cost of SE and SL to start enumeration
!use S1 as starting point and S2 as current optimum
IF(C2.LT.C1) THEN
    C1=C2
    S1=S2
    W1=W2
    FLAG=1-FLAG
ELSE
    C2=C1
    S2=S1
    W2=W1
END IF

!go through all possibilities using a binary counting scheme
S=S1
CHECKSUM=0
DO J=1,2**I-2
    H=1
    !next schedule in binary scheme
    CALL FLIP(H,FLAG,DIFFER,S,S1,CHECKSUM)

    !no need to evaluate if SE or SL differ by just one place,
    !since these schedules are already evaluated
    IF(CHECKSUM.EQ.1.OR.CHECKSUM.EQ.I-1) GOTO 200

    !no need to evaluate if customers have changed order
    !since these schedules are infeasible
    DO H=1,N-1
        IF(S(H).GT.S(H+1)) GOTO 200
    END DO

    !schedule must be evaluated
```

```
        CALL EVALUATE(S,W,C)
        NIT=NIT+1
        IF(C.LT.C2)THEN
            C2=C
            S2=S
            W2=W
        END IF

200 END DO

100 RETURN
    END

    !*******************************

    RECURSIVE SUBROUTINE FLIP(J,FLAG,DIFFER,S,S1,
1       CHECKSUM)
    !flips Jth arrival of current schedule.  If the position
    !is the same as that in the starting schedule, it calls itself
    !recursively to flip the (J+1)st customer

    USE SETSIZE
    INTEGER J,I,S(N+1),S1(N+1),DIFFER(N),CHECKSUM,FLAG

    I=DIFFER(J)
    S(I)=S(I)+1-2*FLAG
    IF(ABS(S1(I)-S(I)).EQ.2) THEN
        S(I)=S1(I)
        CALL FLIP(J+1,FLAG,DIFFER,S,S1,CHECKSUM)
        CHECKSUM=CHECKSUM-1
    ELSE
        CHECKSUM=CHECKSUM+1
    END IF
    RETURN
    END

    !*******************************

    SUBROUTINE EVALUATE(TAU,W,COST)

    !compute cost and waiting time vector of schedule TAU.

    !Value of TAU(N+1) is modified here and then set back to K-1.
    !overtime is the waiting time of a fictitious customer at
    !TAU(N+1), plus time elapsed from OTSLOT to TAU(N), if any.
```

```
!N number of customers
!R number of phases for each customer
!NQ used dimensions of Q
!E EXP(Q*DELTA)
!Q input matrix
!ERROR max error allowed in computation of exp(Q*DELTA)
!TAU arrival vector (schedule), in units of DELTA
!GAMMA prob of each customer showing
!DELTA schedule lattice size
!APHASE number of phases arrived in system so far
!W expected wait of each customer
!OTSLOT slot in which overtime begins
!OTSLOT2 =TAU(N) if TAU(N)>OTSLOT, =OTSLOT otherwise
!OTPOINT onset of overtime

USE SETQDATA
USE SETCOSTDATA
INTEGER TAU(N+1),I,J,DT,APHASE !,DISTFLAG
REAL*8 WAITCOST,COST,PM(LDA),PP(LDA),OVERCOST
REAL*8 WW(N+1),W(N+1)
!SAVE DISTFLAG

PM=0.0D0 !prob vector up to next arrival
PP=0.0D0 !prob vector just after arrival
PM(1)=1.0D0
APHASE=0 !sum of customers' phases so far
W(1)=0.0D0 !individual customers waits, assuming they showed
WAITCOST=0.0D0 !total wait
WW=0.0D+0 !tracking variable -- obsolete

!Set TAU(N+1) to facilitate calculation of overtime
TAU(N+1)=MAX(OTSLOT,K-1)

DO J=1,N
    W(J+1)=0.0D+0
    !place all probability mass beyond APHASE at APHASE+1
    DO I=APHASE+2,NQ
        PM(APHASE+1)=PM(APHASE+1)+PM(I)
        PM(I)=0.0D+0
    END DO

    !push (1-GAMMA) of the exit probability mass to the exit of
    !the next arrival to account for the probability of a no-show.
    PP=PM
```

```
    PP(APHASE+R(J)+1)=(1-GAMMA(J))*PM(APHASE+1)
    PP(APHASE+1)=GAMMA(J)*PM(APHASE+1)

    APHASE=APHASE+R(J)

    !find probability vector at TAU(J+1)
    DT=TAU(J+1)-TAU(J)
    CALL PEXTEND(DT,PP,PM)

    !find W(J+1)
    DO I=1,APHASE
        W(J+1)=W(J+1)+PM(I)*OMEGA(J+1,I)
        IF(I.GT.APHASE-R(J)) WW(J+1)=WW(J+1)+PM(I)*OMEGA(J+1,I)
    END DO

    !GAMMA(J) factor added 25Feb97 to make J's wait be zero if no-show
    WAITCOST=WAITCOST+W(J)*GAMMA(J)*CW(J)

END DO

!if OTPOINT>K-1, then TAU(N+1) is placed at OTSLOT,
!and overtime is W(N+1).
!otherwise, TAU(N+1) is left at K-1,
!and overtime is W(N+1)+(K-1-OTSLOT)*DELTA
OVERCOST = (W(N+1)+MAX(0,TAU(N+1)-OTSLOT)*DELTA)*CW(N+1)
COST=WAITCOST+OVERCOST
!reset TAU(N+1) for use as a bound in FATHOM
TAU(N+1)=K-1
!WRITE(2,*)(TAU(J),J=1,N+1),COST

RETURN
END

!**************************************
```

SUBROUTINE PEXTEND(DT,PP,PM)

```
!finds PM = PP*exp(Q*DELTA*(tau(J+1)-tau(J)))
!accounts for change in state between times
!TAU(J) and TAU(J+1)

!PM(J) state probability vector just before J's arrival
!PP(J) state probability vector just after J's arrival
!DT time step between arrivals
```

```
USE SETQDATA
INTEGER J,I,DT
REAL*8 PP(LDA),PM(LDA),ET(LDA,LDA)

!initialize ET to identity matrix
ET=0.0D0
DO J=1,NQ+1
    ET(J,J)=1.0D0
    PM(J)=0.0D0
END DO

!ET=EXP(Q*DELTA*DT)
DO J=1,DT
    CALL MATMULT(ET,E,NQ+1,LDA)
END DO

DO J=1,NQ+1
    DO I=1,NQ+1
        PM(J)= PM(J)+PP(I)*ET(I,J)
    END DO
END DO

RETURN
END

!**************************************
```

SUBROUTINE GREEDYSEQ(ERR,ITER,NSOFAR)

```
!tests whether greedy algorithm is effective when
!started from two sequences (input).  It compares
!result to global optimum found by exhaustive
!enumeration

!Greedy Algorithm:  From some starting sequence, find
!the pairwise swap that yields the greatest cost
!improvement, if any.  If none, optimum is reached.
!Otherwise, make the swap and repeat the process.

USE SETSEQDATA

INTEGER H,I,J,NSOFAR,SAME(N,N),FLAG,BADALG
INTEGER ITER
CHARACTER*10 TEMPC
REAL*8 ERR
```

```
      BADALG=0
      ITER=0
      DO H=1,1 !index of test starting sequence
5         CALL BUILDSAME(ALPH(NSOFAR),N,SAME)
          FLAG=0
          ITER=ITER+1
          DO I=1,N-1
              DO J=I+1,N !I and J are swap indices

              IF(SAME(I,J).EQ.0) THEN !valid sequence...

                  !TEMPC is ALPH(NSOFAR,*) with I and J swapped
                  DO II=1,N
                      TEMPC=ALPH(NSOFAR)
                  END DO
                  TEMPC(J:J)=ALPH(NSOFAR)(I:I)
                  TEMPC(I:I)=ALPH(NSOFAR)(J:J)


                  !find index of corresponding sequence
                  DO II=1,NSEQ
                      IF(ALPH(II).EQ.TEMPC) GOTO 20
10                END DO
                  WRITE(*,*)'ERROR IN SWAP ROUTINE'
                  WRITE(4,*)'ERROR IN SWAP ROUTINE'
20                CONTINUE !II is now index of swapped sequence

                  IF(C(II).LT.C(NSOFAR)) THEN
                      NSOFAR=II
                      GOTO 5
                  END IF
              END IF !valid sequence...
          END DO !J
      END DO !I

      ERR=(C(NSOFAR)-C(NBEST))/C(NBEST)

30    END DO !H

      RETURN
      END

      !*******************************
```

SUBROUTINE BUILDSAME(TEMPLATE,N,SAME)

```
!generate SAME, where SAME(I,J)=1 if service
!distributions of I and J are the same.

INTEGER I,J,SAME(N,N)
CHARACTER*10 TEMPLATE

SAME=1
DO I=1,N-1
   DO J=I+1,N
      IF(TEMPLATE(I:I).NE.TEMPLATE(J:J)) THEN
         SAME(I,J)=0
         SAME(J,I)=0
      END IF
   END DO !J
END DO !I

RETURN
END

!*****************************
```

SUBROUTINE INPUT

```
!Accepts information from COXINPUT.TXT. Format of this
!file is described in its comments

USE SETQDATA
USE SETCOSTDATA
INTEGER I,J,ERRCODE
CHARACTER A*10
OPEN (1, FILE='coxinput.txt')
ERRCODE=0

READ (1,*) A
READ (1,*) A
READ (1,*) N
IF(N.GT.MAXCUST) THEN
   ERRCODE=1
   GOTO 100
ENDIF

READ (1,*) A
READ (1,*) (R(J),J=1,N)
```

```
      DO J=1,N
         IF(R(J).LT.0.OR.R(J).GT.MAXPHASES) ERRCODE=2
      END DO
      READ (1,*) A
      DO J=1,N
         READ (1,*) (MU(J,I),I=1,R(J))
         DO I=1,R(J)
            IF(MU(J,I).LE.0) ERRCODE=3
         END DO
      END DO

      READ (1,*) A
      READ (1,*) A
      DO J=1,N
         READ (1,*) GAMMA(J),(B(J,I), I=1,R(J)-1)
         IF(GAMMA(J).GT.1.0D0.OR.GAMMA(J).LT.0.0D0) ERRCODE=4
         DO I=1,R(J)-1
            IF(B(J,I).GT.1.0D0.OR.B(J,I).LT.0.0D0) ERRCODE=5
         END DO
      END DO
      GAMMA(N+1)=1.0D0

      READ (1,*) A
      READ (1,*) (CW(J),J=1,N+1)

100   IF(ERRCODE.NE.0) THEN
         SELECT CASE (ERRCODE)
         CASE(1)
            WRITE(*,*)'ERROR READING NUMBER OF CUSTOMERS - MAX EXCEEDED'
         CASE(2)
            WRITE(*,*)'ERROR READING NUMBER OF PHASES - MAX EXCEEDED'
         CASE(3)
            WRITE(*,*)'ERROR READING PHASE RATES'
         CASE(4)
            WRITE(*,*)'ERROR READING SHOW PROBABILITIES'
         CASE(5)
            WRITE(*,*)'ERROR READING PHASE PROBABILITIES'
         CASE(6)
            WRITE(*,*)'ERROR READING DELTA'
         CASE DEFAULT
            WRITE(*,*)'UNSPECIFIED ERROR'
         END SELECT
      END IF
      CLOSE (1)
      RETURN
```

```
END

!*****************************
```

SUBROUTINE BUILDQ

```
!Builds Q matrix from data in COXINPUT.TXT. Returns NQ
!Also calls OMEVAL to build conditional wait matrix OMEGA
!modified 26feb97 to incorporate show prob into Q

USE SETQDATA
INTEGER I,J
Q=0.0D0
NQ=0

!LDA max size of Q.
!NQ index of current row of matrix.  ends as size of Q. Output.
!MU transition rate of each phase
!B routing probability to next phase.
!note:  B(H,0) is show rate of customer H. Not accounted for in Q.

!determine Q
DO J=1,N
   DO I=1,R(J)-1
      Q(NQ+I,NQ+I)=-MU(J,I)*DELTA
      Q(NQ+I,NQ+I+1)=B(J,I)*MU(J,I)*DELTA
      IF(J.EQ.N) THEN
         Q(NQ+I,NQ+R(J)+1)=(1-B(J,I))*MU(J,I)*DELTA
      ELSE
         Q(NQ+I,NQ+R(J)+1)=GAMMA(J+1)*(1-B(J,I))*MU(J,I)*DELTA
         Q(NQ+I,NQ+R(J)+R(J+1)+1)=
1        (1-GAMMA(J+1))*(1-B(J,I))*MU(J,I)*DELTA
      END IF
   END DO
   NQ=NQ+R(J)
   Q(NQ,NQ)=-MU(J,R(J))*DELTA
   Q(NQ,NQ+1)=MU(J,R(J))*DELTA
   DO I=1,NQ
      IF(J.NE.N)THEN
         Q(I,NQ+1+R(J+1))=(1-GAMMA(J+1))*Q(I,NQ+1)
         Q(I,NQ+1)=GAMMA(J+1)*Q(I,NQ+1)
      END IF
   END DO
END DO
```

```
!append a last row of zeros to Q (exit state)
NQ=NQ+1
RETURN
END

!**********************************
```

SUBROUTINE OMEGAEVAL

```
!evaluates conditional waiting matrix OMEGA.
!OMEGA(J,I) is the expected waiting time for the Jth
!customer, given the current phase is I, and assuming
!all customers show and are immediately available

USE SETQDATA
INTEGER J,IN,IM,IQ
REAL*8 SVC(MAXCUST+1,LDA)
OMEGA=0.0D0
SVC=0.0D0
!define SVC(J,I), the expected service of customer J, given
!the system is in the customer's Ith phase of service
DO J=1,N
   SVC(J,R(J))=1/MU(J,R(J))
   DO I=R(J)-1,1,-1
      SVC(J,I)=SVC(J,I+1)*B(J,I)+1/MU(J,I)
   END DO
END DO

!find OMEGA(J,IQ), J's expected wait, given the current state is IQ
IQ=0
DO IN=1,N ! customer index
   DO IM=1,R(IN) ! customer's stage index
      IQ=IQ+1 !  current state index
      OMEGA(IN+1,IQ)=SVC(IN,IM) ! add partial svc of current customer
      DO J=IN+2,N+1 !  customer #
         !add full (possible) svc of other customers (recursively)
         OMEGA(J,IQ)=OMEGA(J-1,IQ)+GAMMA(J-1)*SVC(J-1,1)
      END DO
   END DO
END DO

RETURN
END
```

```
!*********************************
```

SUBROUTINE EXP(ERROR)

```
!Employs a scale-and square algorithm to create a matrix that has
!a smaller determinant, using a Taylor series approximation to obtain
!exp(Q/M), then taking that matrix to the mth power to get exp(Q).
!NP is the smallest power of 2 that ensures det(Q*DELTA/NP) < 1.
!Stopping criteria are max # terms in Taylor series (100) or Liou's
!criterion, based on the 2-norm (Proc IEEE, 54(1966)20-23), whichever
!is satisfied first.
!Only the exponential of the P matrix (Q matrix minus the last row
!and column) is calculated, after which the last row and column are
!computed and appended, allowing slight savings in time and accuracy.
!  Q input matrix
!  QM Q scaled by F
!  QT current term of Taylor series of QM
!  F scaling factor
!  NORM Frobenius norm.  Should actually be 2-norm, but F-norm
!  is always larger, so this is conservative
!  ERROR allowable error in scaled matrix values

USE SETQDATA
INTEGER I,J,H
REAL*8 QM(LDA,LDA),NORM,QT(LDA,LDA),FACT,ERROR
!initialize
DO I=1,NQ-1
   DO J=1,NQ-1
      QM(I,J)=Q(I,J)
      QT(I,J)=0.0D0
      E(I,J)=0.0D0
   END DO
   QT(I,I)=1.0D0
   E(I,I)=1.0D0
END DO

!find the Frobenius norm
NORM=0.D0
DO I=1,NQ
   DO J=1,NQ
      NORM=NORM+Q(I,J)**2
   END DO
```

```
      END DO
      NORM=SQRT(NORM)

      !NP is min # halvings needed to scale DET(P).
      !F is smallest power of 2 larger than F-norm of Q
      NP=MAX(0,INT(LOG(ABS(NORM))/LOG(2.0D0)+1))
      F=2.0D0**DBLE(NP)
      NORM=NORM/F

      !scale QM
      DO I=1,NQ-1
         DO J=1,NQ-1
            QM(I,J)=QM(I,J)/F
         END DO
      END DO

      FACT=1.0D0
      DO H=1,100
      !  WRITE(1,*)
      !  WRITE(1,*)'TERM ',H
         CALL MATMULT(QT,QM,NQ,LDA)
         DO I=1,NQ-1
            DO J=1,NQ-1
               QT(I,J)=QT(I,J)/H
               E(I,J)=E(I,J)+QT(I,J)
            END DO
      !     WRITE(1,10)(QT(I,J),J=1,NQ)
         END DO
         FACT=FACT*(H+1)
         !Standish's stopping condition
         IF((NORM*2)**H/(2*FACT).LT.ERROR) GOTO 20

         !Liou's stopping condition
         IF(NORM*(H+2)/((H+2-NORM)*FACT).LT.ERROR) GOTO 20
      END DO
      WRITE(2,*)
      WRITE(2,*)'Warning:  exp(Q*DELTA) may not be accurate.'
      WRITE(2,*) '100 terms of the Taylor series were used without achieving
     1   the stopping condition'
10    FORMAT(<NQ>E9.2)

      !E is now exp of the scaled P matrix.
      !First, unscale by squaring E NP times.
20    DO I=1,NP
         CALL MATMULT(E,E,NQ,LDA)
```

```
END DO

!now add last column and last row to complete exp(Q*DELTA)
DO I=1,NQ-1
   E(I,NQ)=1.0D0
   DO J=1,NQ-1
      E(I,NQ)=E(I,NQ)-E(I,J)
   END DO
   E(NQ,I)=0.0D0
END DO
E(NQ,NQ)=1.0D0

RETURN
END

!*************************
```

SUBROUTINE MATMULT(A,B,N,LDA)

```
!calculates A=A*B, where A,B are upper triangular and real.
!N is order of the used matrices.
!Use of triangular mult requires only 1/6 the number of flops
!as does full mult (see Golub+Van Loan,p 18)

INTEGER I,J,H,N
REAL*8 A(LDA,LDA),B(LDA,LDA),C(LDA,LDA)
C=0.0D0

DO I=1,N
   DO J=I,N
      !C(I,J) is product of Ith row of A and Jth column of B
      DO H=I,J
         C(I,J)= C(I,J)+A(I,H)*B(H,J)
      END DO
   END DO
END DO

A=C

RETURN
END

!*************************
```

SUBROUTINE FPFLAW

```
!checks for original Pentium floating point bug
!borrowed from Microsoft.
!cf:  FORTRAN PowerStation Programmer's Guide, pp550-1

REAL OP1,OP2
COMMON /DIVIDECHECK/ OP1,OP2
DATA OP1 /3145727.0/,OP2 /4195835.0/
IF(OP2/OP1.LE.1.3338) THEN
   WRITE(*,*)' WARNING: PENTIUM FLAW DETECTED. THERE IS A SMALL'
   WRITE(*,*)' CHANCE THE PROGRAM WILL GIVE INCORRECT RESULTS'
   WRITE(2,*)' WARNING: PENTIUM FLAW DETECTED. THERE IS A SMALL'
   WRITE(2,*)' CHANCE THE PROGRAM WILL GIVE INCORRECT RESULTS'
   WRITE(*,*)' CONTACT INTEL AT 1-800-628-8686 FOR MORE INFORMATION'
   WRITE(2,*)' CONTACT INTEL AT 1-800-628-8686 FOR MORE INFORMATION'
END IF

RETURN
END
```

H.2 Input Files

This section describes the use of and gives examples of the input files used, as well as the program used to generate the SEQUENCE file.

- DISTRIB (page 227): Input file. Tabulates Coxian parameters for a number of 3-moment sets (mean-normalized). Accessed by HCSEARCH.
- COXINPUT (page 227): Input file for Coxian parameters. Not used in this particular version, since Coxian parameters were generated randomly. Accessed by subroutine INPUT.
- SEQUENCE (page 228): Input file. Enumerates all the sequences to be evaluated in the search for the global optimum. Accessed by HCSEARCH.
- PERMUTE (page 228): Program to generate permutations of customers. It produces the file SEQUENCE in cases where all permutations are to be tested.

Input file DISTRIB

```
phi2 , phi3, #Erlang phases, Coxian rate, Erlang rates, trans prob
assumes first moment is 1.0
4.0         6.0         0          1.0          1.0          0.0
1.5         3.0         1          2.0          2.0          1.0
1.25        1.87        3          4.0          4.0          1.0
3.00        11.3        3          105.2        1.329        0.4388
5.0         31.3        3          425.5        0.7989       0.2657
3.00        67.4        1          1.029        0.05397      0.00154
5.0         188.0       1          1.095        0.05468      0.00475
0.0         0.0         0          0.0          0.0          0.0

!************************************
```

Input file COXINPUT

```
INPUT DATA. UNFORMATTED, BUT SOME LINES ARE RESERVED FOR COMMENTS
# Customers
5
# phases for each customer
8          8          8          8          8
phase rates mu(j,k), cust indexed by rows(j), phases indexed by columns(k)
0.360 0.360 0.360 0.360 0.360 0.360 0.360 0.360
0.360 0.360 0.360 0.360 0.360 0.360 0.360 0.360
0.308 0.308 0.308 0.308 0.308 0.308 0.308 0.308
0.308 0.308 0.308 0.308 0.308 0.308 0.308 0.308
0.308 0.308 0.308 0.308 0.308 0.308 0.308 0.308
transition probs b(j,k), cust indexed by rows, phases indexed by columns.
b(j,0) is gamma(j), the show probability
0.95 0.984 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
0.95 0.984 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
0.92 0.982 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
0.92 0.982 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
0.92 0.982 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
cost coefficients 2 through N+1
1.0D+00 1.0D+00 1.0D+00 1.0D+00 1.0D+00

!************************************
```

Input file SEQUENCE

```
ALL 3-CUSTOMER SEQUENCES
1       2       3
1       3       2
2       1       3
2       3       1
3       1       2
3       2       1
0       0       0
```

```
!**************************************
```

PROGRAM PERMUTE

```
      !program to create a file of permutations
      !for use in validating sequencing algorithms
      !in program HCSEARCH

      INTEGER I,N,PERM(400000,9),PERM1(400000,9),NPERM
      CHARACTER*8 ALPH
      OPEN(1,FILE='OUT')

      ALPH='123456789'  !put items to be permuted here
      WRITE(*,*)'Input number of objects to be permuted:'
      READ(*,*)N
      PERM1(1,1)=1
      NPERM=1
      I=1
      CALL PERMADD(N,I,NPERM,PERM1,PERM)

      I=1
      WRITE(1,10)'All the permutations of ',N,' customers'
10    FORMAT(A24,I2,A10)
      DO WHILE(PERM(I,1).GT.0)
         WRITE(1,*)(ALPH(PERM(I,J):PERM(I,J)),' ',J=1,N)
         I=I+1
      END DO

      !append a row of zeros for use in HCSEARCH
      WRITE(1,*) ('0 ',I=1,N)
      WRITE(*,*)I-1,'TOTAL PERMUTATIONS CALCULATED'

      CLOSE(1)
      END
```

```
!**************************

      RECURSIVE SUBROUTINE PERMADD(N,I,NPERM,PERM1,PERM)
      !takes each of the I! permutations of I customers
      !and inserts the (I+1)st customer at every possible
      !point to create all permutations of I+1 customers

      INTEGER COUNT,N,I,NPERM,PERM(400000,9),PERM1(400000,9),J,K
      COUNT=0
      I=I+1
      DO J=1,NPERM
         DO K=1,I
            COUNT=COUNT+1
            DO M=1,K-1
               PERM(COUNT,M)=PERM1(J,M)
            END DO
            PERM(COUNT,K)=I
            DO M=K+1,I
               PERM(COUNT,M)=PERM1(J,M-1)
            END DO
         END DO
      END DO
      DO J=1,COUNT
         DO K=1,I
            PERM1(J,K)=PERM(J,K)
         END DO
      END DO
      IF(I.LT.N) CALL PERMADD(N,I,COUNT,PERM1,PERM)

      RETURN
      END
```

H.3 Alternative Matrix Exponentiation Routines

The following routines are alternatives to the matrix exponentiation routine EXP (page 223 in the previous section). The algorithms provide mathematically exact results, but when they are applied using floating-point arithmetic, they may produce substandard results for the exponentiations required in schedule evaluations. They are included here only to support the discussion in Appendix G.

- CAYHAM (page 230): Cayley-Hamilton algorithm discussed in Section G.3. Calls subroutine EIGENVAL (page 233).
- PARLETT (page 234): Parlett's algorithm discussed in Section G.5.

SUBROUTINE CAYHAM(Q2,E2,LDA,NQ,EMAX)

```
      !Finds the exponential of an NxN matrix using Cayley-Hamilton
      !theorem.  Allows eigenvalues to be complex, but that capability
      !isn't needed for this dissertation.  The matrix is required to be
      !real here, but that requirement can be relaxed by changing types
      !in all routines.  The program can also be used to find other
      !well-defined functions of a matrix, simply by changing the
      !definition of B.

      !Errors in this approach are almost entirely due to floating point
      !truncation of very large contributing terms which nearly cancel
      !each other (Van Loan's ''catastrophic cancellation").
      !Therefore, error is estimated by comparing the size of
      !the absolute value of the largest contributing term to the final
      !result in each element of the computed exp(Q). Double precision
      !arithmetic allows for only 15 decimal places of accuracy, so if
      !the ratio is larger than 1E13, there is the possibility that the
      !result has fewer than two decimal places of accuracy.

!     EIGVAL              the eigenvalues of Q
!     EIG                 the distinct eigenvalues of Q
!     MULT                the multiplicity of each eigenvalue in EIG
!     NEIG                the number of elements in EIG
!     LDA                 the dimensions of the matrix storage spaces
!     NEQ                 used dimensions of matrices
!     E                   EXP(Q)
!     EMAX                max of the abs value of the terms E comprises
!     ETERMS              the matrix of abs values of the terms E comprises

      !Microsoft-specific:  invokes IMSL link
      USE MSIMSL

      INTEGER I,J,K,UPTRI,NEIG,MULT(LDA),FACT(0:LDA),IPATH,NQ
      COMPLEX(8) EIGVAL(LDA),EIG(LDA),A(LDA),C(LDA,LDA),B(LDA), E(LDA,LDA)
      REAL*8 Q(LDA,LDA), QP(LDA,LDA), TEMP ,TEMP2

      REAL*8 E2(LDA,LDA),Q2(LDA,LDA),EMAX(LDA,LDA)
```

```
      !initialize
      DO I=1,LDA
         DO J=1,LDA
            Q(I,J)=Q2(I,J)
            C(I,J)=0.00D+00
            E(I,J)=0.00D+00
            EMAX(I,J)=0.00D+00
            QP(I,J)=0.0D+00
         END DO
      END DO

      !tell EIGENVAL routine that Q is upper triangular.  get eigenvalues
      UPTRI=1
      CALL EIGENVAL(Q,NQ,LDA,UPTRI,EIGVAL)

      !determine list of distinct eigenvectors and their multiplicities
      EIG(1)=EIGVAL(1)
      NEIG=1
      MULT(1)=1
      DO I=2,NQ
         DO J=1,NEIG
            IF(EIGVAL(I).EQ.EIG(J)) THEN
               MULT(J)=MULT(J)+1
               GOTO 10
            ENDIF
         END DO
         NEIG=NEIG+1
         MULT(NEIG)=1
         EIG(NEIG)=EIGVAL(I)
10    END DO

      !calculate factorials needed
      FACT(0)=1
      DO I=1,NQ
         FACT(I)=FACT(I-1)*I
      END DO

      !calculate C and B in B=CA, where A are the polynomial coefficients
      !in exp(Q)=A(0)+A(1)*Q**1+A(2)*Q**2+...+A(N)*Q**N
      !A and B may be complex if q isn't real or if it isn't triangular
      !NEQ=0
      DO I=1,NEIG
         DO K=1,MULT(I)
            NEQ=NEQ+1
```

```
      DO J=K-1,NQ-1
         C(NEQ,J+1)=EIG(I)**(J-K+1)*FACT(J)/FACT(J-K+1)
      END DO
      !if another (well-defined) function of Q other than exp(Q) is
      !to be calculated, only the following statement need be changed
      !to reflect the derivative of order K-1 with respect to Q of
      !that function
      B(NEQ)=EXP(EIG(I))
      !WRITE(1,*) 'B(',NEQ,')=',B(NEQ)
      !WRITE(1,*) (C(NEQ,J),J=1,NQ)
      !WRITE(1,*)
   END DO
END DO

!solve B=C*A for vector A using IMSL routine
IPATH=1
CALL DLSACG(NQ,C,LDA,B,IPATH,A)
!WRITE(1,*)'A Coefficients:'
!WRITE(1,*)(A(I),I=1,NQ)

!continue to add polynomial terms to E
!inefficient, but we need to track magnitudes of each contribution
!to each element to ensure the range is not too great for accuracy
DO I=1,NQ
   E(I,I)=A(1)
   QP(I,I)=1.00D+00
END DO

DO I=1,NQ-1
   CALL MATMULT(QP,Q,NQ,LDA)
   DO J=1,NQ
      DO K=1,NQ
         TEMP=ABS(A(I+1))*QP(J,K)
         EMAX(J,K)=MAX(TEMP,EMAX(J,K))
         E(J,K)=E(J,K)+TEMP
         IF(J.EQ.1.AND.K.EQ.4)WRITE(1,*)TEMP,E(J,K)
      END DO
   END DO
END DO

!if error analysis not needed, the following is more efficient
!it requires a type change in MATMULT, though
!calculate E=A(0)+A(1)E+...+A(NQ)E**NQ
!recursively calculates E=E*Q+A(J)*IDENTITY
!DO I=NQ-1,1,-1
```

```
         !calculates E=E*Q
         !CALL MATMULT(E,Q,NQ,LDA)
         !DO J=1,NQ
            !E(J,J)=E(J,J)+A(I)
         !END DO
      !END DO

20    FORMAT(<NQ>E10.2)

      !use ratio of E to EMAX to see if there were massive cancellations
      TEMP=1.E30
      WRITE(1,*)
      WRITE(1,*)
      DO I=1,NQ
         DO J=1,NQ
            TEMP2=ABS(E(I,J)+1.E-30)/ABS(EMAX(I,J)+1.E-30)
            TEMP=MIN(TEMP,TEMP2)
            IF(TEMP2.LE.1E-13) THEN
               WRITE(1,*)I,J,' Some elements of EXP(Q) had a very large
1                 cancellation.'
               WRITE(1,*)'There may be an accuracy problem as a result.'
               GOTO 40
            END IF
         END DO
      END DO

      !pass back REAL*4 representation of E
40    DO I=1,NQ
         DO J=1,NQ
            E2(I,J)=REAL(E(I,J))
         END DO
      END DO

      RETURN
      END

!************************************

      SUBROUTINE EIGENVAL(Q,NQ,LDA,UPTRI,EIGVAL)
      !Finds eigenvalues of Q, either using IMSL routine or, in the cases
      !of interest to this dissertation, since Q is upper-triangular,
      !simply returns diagonal of Q.

      INTEGER LDA,I,UPTRI,NQ
      REAL*8 Q(LDA,LDA)
```

```
COMPLEX(8) EIGVAL(LDA)

!find eigenvalues
IF(UPTRI.EQ.1) THEN
   DO I=1,LDA
      EIGVAL(I)=Q(I,I)
   END DO
   ELSE
      CALL DEVLRG(NQ,Q,LDA,EIGVAL)
END IF

!WRITE(*,*)'Eigenvalues:  ',(EIGVAL(I),I=1,N)

RETURN
END
```

SUBROUTINE PARLETT(Q2,E2,LDA,NQ,ERROR)

```
!Employs Parlett's method to obtain exp(Q/m).  Since that method does
!not allow for confluent eigenvalues, a small amount (ERROR) is added
!or subtracted from each confluent eigenvalue to ensure they are
!distinct.  If there are N confluent eigenvalues, this routine causes
!them to differ by between 0 and INT(N/2)*2E. Once the
!eigenvalues are adjusted, the entire row is scaled accordingly.

!     LDA          the dimensions of the matrix storage spaces
!     NQ           used dimensions of matrices
!     E            EXP(Q)
!     Q            input matrix
!     ERROR        adjustments to eigenvalues
!     ICOUNT       counts number of eigenvalues identical to current test
!     FLAG         flags possibly catastrophic cancellations
!     TMAX         max term of the element of E currently being calculated

INTEGER I,J,NQ,LDA,K,FLAG
REAL*8 Q(NQ,NQ),E(NQ,NQ),TMAX, TERM, ERROR
REAL*4 Q2(LDA,LDA),E2(LDA,LDA)

!initialize
FLAG=0
DO I=1,NQ-1
  DO J=1,NQ-1
    Q(I,J)=Q2(I,J)
```

```
          E(I,J)=0.0E+00
        END DO
      END DO

      DO I=1,NQ-2
        ICOUNT=1
        DO J=I+1,NQ-1
          IF(Q(I,I).EQ.Q(J,J))THEN
            ICOUNT=ICOUNT+1
            C=1+(-1)**ICOUNT*INT(ICOUNT/2)*ERROR
            DO K=J,NQ-1
              Q(J,K)=Q(J,K)*C
            END DO
          END IF
        END DO
      END DO

      DO I=1,NQ-2
        DO J=I+1,NQ-1
          IF(Q(I,I).EQ.Q(J,J))THEN
            WRITE(1,*)'This is one of the unlikely cases in which the
     1      eigenvalue dithering routine produced two '
            WRITE(1,*)'or more confluent eigenvalues.  Suggest you '
     1      restart the routine after changing ERROR slightly.'
            STOP
          END IF
        END DO
      END DO

      WRITE(1,*)'P:'
      DO J=1, NQ-1
        WRITE(1,10)(Q(J,I), I=1,NQ-1)
      END DO
10    FORMAT(<NQ-1>E13.6)

      !now that eigenvalues are distinct, we can use Parlett's algorithm
      DO I=1,NQ-1
        E(I,I)=EXP(Q(I,I))
      END DO

      DO I=1,NQ-2 !# diagonals from main diagonal
        DO J=1,NQ-1-I !# row.  # column is I+J
          E(J,I+J)=Q(J,I+J)*(E(I+J,I+J)-E(J,J))
          TMAX=E(I,J)
          DO K=J+1,I+J-1
```

```
          TERM=Q(J,K)*E(K,I+J)-E(J,K)*Q(K,I+J)
          TMAX=MAX(TMAX,ABS(TERM))
          E(J,I+J)= E(J,I+J)+TERM
        END DO
        IF(TMAX/E(J,I+J).GT.1.0E+13) FLAG=1
        E(J,I+J)=E(J,I+J)/(Q(I+J,I+J)-Q(J,J))
      END DO
    END DO

    !establish last row and column of E
    DO I=1,NQ-1
      E(I,NQ)=1.0D0
      DO J=I,NQ-1
        E(I,NQ)=E(I,NQ)-E(I,J)
      END DO
      E(NQ,I)=0.0D+00
    END DO
    E(NQ,NQ)=1.0D0

    !must return E as REAL*4
    E2(I,J)=E(I,J)

    IF(FLAG.EQ.1) THEN
      WRITE(1,*)
      WRITE(1,*)'Warning:  one or more elements of EXP(Q) had potentially
   1    catastrophic cancellation.  '
      WRITE(1,*) Accuracy of EXP(Q) is questionable.  Try re-running with
   1    larger ERROR if feasible.'
    END IF

    RETURN
    END
```

H.4 Moment Matching Routine

This Microsoft Excel$_{TM}$ spreadsheet is used to determine the parameters r, w, and b for the Cox-plus-Erlang-r distribution that matches the desired first three (noncentral) moments, m_1, m_2, and m_3, as discussed in Section F.9. It uses Excel's resident NLP to perform a line search to find the best value of w. The other parameters are determined analytically.

	A	B	C
1		Find a Cox+Erlang-r distribution which	
2		matches the first 3 moments	
3			
4		Change only these moments:	
5	m1	1	
6	m2	3	
7	m3	10	
8			
9	phi2	=B6/B5**2	
10	phi3	=B7/B5**3	
11		=IF(B10/B9**2.ge.1"PROBLEM IS FEASIBLE","NOT FEASIBLE!")	
12			
13	1	=IF((A13+2)/(A13+1).lt.B10/B9**2,1,0)	=A13*B13
14	2	=IF(AND((A14+2)/(A14+1).lt.B10/B9**2,(A14+1)/(A14).ge.B10/B9**2),1,0)	=A14*B14
15	3	=IF(AND((A15+2)/(A15+1).lt.B10/B9**2,(A15+1)/(A15).ge.B10/B9**2),1,0)	=A15*B15
16	4	=IF(AND((A16+2)/(A16+1).lt.B10/B9**2,(A16+1)/(A16).ge.B10/B9**2),1,0)	=A16*B16
17	5	=IF(AND((A17+2)/(A17+1).lt.B10/B9**2,(A17+1)/(A17).ge.B10/B9**2),1,0)	=A17*B17
18	6	=IF(AND((A18+2)/(A18+1).lt.B10/B9**2,(A18+1)/(A18).ge.B10/B9**2),1,0)	=A18*B18
19	7	=IF(AND((A19+2)/(A19+1).lt.B10/B9**2,(A19+1)/(A19).ge.B10/B9**2),1,0)	=A19*B19
20	8	=IF(AND((A20+2)/(A20+1).lt.B10/B9**2,(A20+1)/(A20).ge.B10/B9**2),1,0)	=A20*B20
21	9	=IF(AND((A21+2)/(A21+1).lt.B10/B9**2,(A21+1)/(A21).ge.B10/B9**2),1,0)	=A21*B21
22	10	=IF(AND((A22+2)/(A22+1).lt.B10/B9**2,(A22+1)/(A22).ge.B10/B9**2),1,0)	=A22*B22
23			
24			
25		To obtain desired coefficients w and b, use Excel's	
26		solver to minimize object wrt w. You may have	
27		to change the sign of zeta to get a feasible answer	
28			
29		=IF(B31=0,"whoa! r must be larger than expected. Expand table"," ")	
30	w	0.1	
31	rr	=SUM(C13:C22)	
32	zeta	=-SQRT((rr+1)**2+4*w*(rr+1)*(1-phi2)+4*w**2)	
33	bb	=(2*w*(1-phi2)+rr+1+zeta)/(2*phi2*rr)	
34	phi3	=(6*w**3+6*bb*rr*w**2+3*bb*rr**2*w +3*bb*rr*w+bb*rr**3+3*bb*rr**2+2*bb*rr)/(w+bb*rr)**3	
35	object	=(B34-B10)**2	

H.5 Scheduling Simulation Code

The following FORTRAN 90 code was used in Appendix E to simulate the assignment of a combination to each day. It is assumed that six customers request appointments each day, each customer belonging to one of three classes. Arrival probabilities are specified for each class. For each day, a combination must be chosen. A combination allots six appointments on a given day, each one being dedicated to one class of customer. The measures of merit of such a system include the average system cost, the number of unfillable appointments, the average delay between patient arrival and appointment, and the number of unacceptable delays. The system is complex enough to require simulation for all but the simplest cases.

PROGRAM COMBINATION

```
!NOW: current day of customer to be scheduled
!NOW2:  nonvolatile copy of NOW
!DAYNOW: what day it is
!LASTFIXED: the last day of schedules already fixed
!POSS: the possible schedules for each day
!COST: cost of each poss
!SCHED: actual schedule each day - volatile
!SCHED2:  nonvolatile copy of SCHED
!WAIT(I): number of customers who had to wait I for appointment
!OFF(I): how much excess schedule capacity for customer I
!SCORE: current rating of each possible candidate combination

INTEGER NOW(3),DAYNOW,SCHED(1000,3),SUM,NOW2(1000,3)
INTEGER POSS(5,3),LASTFIXED, TOTAL(3),OFF(3),DAY,TCOST
INTEGER SCHED2(1000),WAIT(0:25),NUMDAYS,TEMP
REAL R,COST(5),SCORE(5),MAXWAIT
OPEN(1,FILE='COMBIN.DAT')
OPEN(2,FILE='OUT')

WRITE(*,*)'RANDOM SEED:'
READ(*,*)J
WRITE(2,*)'RANDOM SEED: ',J
R=RAND(J)

WRITE(*,*)'How many days to schedule?'
```

```
      READ(*,*)NUMDAYS
      LASTFIXED=0
      WAIT=0

      !read in combination candidates
      DO J=1,5
         READ(1,*)(POSS(J,I),I=1,3),COST(J)
      END DO

      DO DAYNOW=1,NUMDAYS
         NOW=0
         DO J=1,6 !6 new customers in
            R=RAND(0)
            IF(R.LT.0.237) THEN
               NOW(1)=NOW(1)+1
            ELSE IF (R.LT.0.597) THEN
               NOW(2)=NOW(2)+1
            ELSE
               NOW(3)=NOW(3)+1
            END IF
         END DO !J
         DO J=1,3
            NOW2(DAYNOW,J)=NOW(J)
         END DO
         TOTAL=TOTAL+NOW

         !at each iteration, schedule all you can
10       OFF=0
         DO DAY=DAYNOW,LASTFIXED
            MAXWAIT=0.0
            DO J=1,3
               REDUCE=MIN(FLOAT(NOW(J)),FLOAT(SCHED(DAY,J)))
               NOW(J)=NOW(J)-REDUCE
               SCHED(DAY,J)=SCHED(DAY,J)-REDUCE
               WAIT(LASTFIXED-DAYNOW)=WAIT(DAY-DAYNOW)+REDUCE
               MAXWAIT=MAX(MAXWAIT,FLOAT(DAY-DAYNOW))
               OFF(J)=OFF(J)+SCHED(DAY,J)
            END DO
         END DO

         !All are scheduled for current day.  Go to next day.
         IF(NOW(1)+NOW(2)+NOW(3).EQ.0)GOTO 30

         !All are not scheduled.  Choose the schedule for day LASTFIXED+1
         SCORE=0
```

```
         TEMP=10000
         DO J=1,5
            DO I=1,3
               SCORE(J)=SCORE(J)+3*MAX(0.0,FLOAT(NOW(I)-POSS(J,I)))
               SCORE(J)=SCORE(J)+OFF(I)+MAX(0.0,FLOAT(POSS(J,I)-NOW(I)))
            END DO
            SCORE(J)=SCORE(J)
            IF(SCORE(J).LT.TEMP)THEN
               PICK=J
               TEMP=SCORE(J)
            END IF
         END DO

         TCOST=TCOST+COST(PICK) !increment total cost
20       LASTFIXED=LASTFIXED+1
         DO J=1,3 !implement schedule number PICK for day LASTFIXED
            SCHED(LASTFIXED,J)=POSS(PICK,J)
         END DO
         SCHED2(LASTFIXED)=PICK
         GOTO 10

30    END DO !DAYNOW

      !Done.  Record results.
      WRITE(2,*)
      WRITE(2,*) 'TOTAL ARRIVALS: ',(TOTAL(I),I=1,3)
      WRITE(2,*)
      WRITE(2,*)' DAY ARRIVALS SCHEDULE'
      NOW=0
      DO J=1,NUMDAYS
         WRITE(2,40)J,' ',(NOW2(J,I),I=1,3),' ', (SCHED2(J))
40       FORMAT(I3,A4,3I3,A4,I3)
         DO I=1,3
            NOW(I)=NOW(I)+SCHED(J,I)
         END DO
      END DO
      WRITE(2,*)
      WRITE(2,*)'EMPTY SLOTS:',(NOW(I),I=1,3)

      WRITE(2,*)
      WRITE(2,*) 'DAYS TO WAIT #CUST'
      WRITE(*,*) 'DAYS TO WAIT #CUST'
      SUM=0
      DO I=0,25
         WRITE(2,*)I,WAIT(I)
```

```
         WRITE(*,*)I,WAIT(I)
         IF(WAIT(I).GT.0)TEMP=I
         IF(I.GT.5)SUM=SUM+WAIT(I)
      END DO

      WRITE(*,*) 'Max wait was',TEMP
      WRITE(*,*)'Total over was',SUM,' OR',
     1   100.0*FLOAT(SUM)/6.0/FLOAT(NUMDAYS),'%'

      WRITE(*,*)
      WRITE(*,*)'Cost per day was ',FLOAT(TCOST)/FLOAT(NUMDAYS)

      CLOSE(1)
      CLOSE(2)
      END

!************************************
```

Input file COMBIN.DAT

```
1    3    2    13.86
2    1    3    12.01
1    4    1    11.24
3    3    0    4.21
0    0    6    38.6
```

Bibliography

1. Agnihothri, S. R. and P. F. Taylor. "Staffing a Centralized Appointment Scheduling Department in Lourdes Hospital," *Interfaces*, *21*(5):1–11 (1991).
2. Aldous, D. and L. Shepp. "The Least Variable Phase Type Distribution is Erlang," *Communications in Statistics - Stochastic Models*, *3*(3):467–473 (1987).
3. Altiok, T. "On the Phase-Type Approximations of General Distributions," *IIE Transactions*, *17*(2):110–116 (1985).
4. Apostol, T. M. *Mathematical Analysis*. Reading: Addison-Wesley, 1974.
5. Asmussen, S., O. Nerman, and M. Olsson. *Fitting Phase-Type Distribution Via the ME Algorithm*. Technical Report 1994:23, ISSN 0347-2809, Chalmers University of Technology and The University of Göteborg, 1994.
6. Asmussen, S., O. Nerman, and M. Olsson. "Fitting Phase-type Distributions via the EM Algorithm," *Scandinavian Journal of Statistics*, *23*:419–441 (1996).
7. Bailey, N. T. J. "A Study of Queues and Appointment Systems in Hospital Outpatient Departments, with Special Reference to Waiting Times," *Journal of the Royal Statistical Society , Series B*, *14*(2):185–199 (January 1952).
8. Bailey, N. T. J. "Queueing for Medical Care," *Applied Statistics*, *3*:137–45 (1954).
9. Barnoon, S. and H. Wolfe. "Scheduling a Multiple Operating Room System: A Simulation Approach," *Health Services Research*, *3*:272–285 (1968).
10. Beasley, J. E., M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson. "Dynamically Scheduling Aircraft Landings - The Displacement Problem." The Management School, Imperial College, London SW7 2AZ, England, December 1995.
11. Beasley, J. E., M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson. "Scheduling Aircraft Landings - The Static Case." The Management School, Imperial College, London SW7 2AZ, England, December 1995.
12. Belden, D. L., R. K. Hall, and R. J. Quayle. *Outpatient Scheduling - A Simulation Approach*. Unpublished report, Wright-Patterson AFB OH: School of Engineering, Air Force Institute of Technology (AU), February 1972.
13. Beleny, C., Lt Col, USAF, Chief of Wright-Patterson Air Force Base Primary Care Clinic. Meeting, 14 November 1996.
14. Bell, C. E. "Turning Off a Server With Customers Present: Is This Any Way to Run an M/M/c Queue With Removable Servers?," *Operations Research*, *23*:571–574 (1975).

15. Borgwardt, K. H. "Some Distribution Independent Results About the Asymptotic Order of the Average Number of Pivot Steps in the Simplex Method," *Mathematics of Operations Research*, *7*(3):441–462 (1982).

16. Bronson, R. *Theory and Problems of Matrix Operations*. New York: McGraw-Hill, 1989.

17. Bryant, V. *Metric Spaces; Iteration and Application*. Cambridge: Cambridge University Press, 1985.

18. Cao, B. and F. Glover. "Tabu Search and Ejection Chains - Application to a Node Weighted Version of the Cardinality-Constrained TSP," *Management Science*, *43*(7):908–921 (1997).

19. Chang, Cheng-Shang. "A New Ordering for Stochastic Majorization: Theory and Application," *Advances in Applied Probability*, *24*:604–634 (1992).

20. Chang, Cheng-Shang and D. Yao. "Rearrangement, Majorization, and Stochastic Scheduling," *Mathematics of Operations Research*, *18*(3):658–684 (1993).

21. Charnetski, J. "Scheduling Operating Room Surgical Procedures With Early and Late Completion Penalty Costs," *Journal of Operations Management*, *5*(1):91–102 (1984).

22. Churchman, C. W., R. L. Ackoff, and E. L. Arnoff. *Introduction to Operations Research*. New York: John Wiley & Sons, 1957.

23. Colbert, Karen. Personal correspondence. Lahey Computer Systems, Inc., 19–23 September 1996.

24. Conway, R. W. and W. L. Maxwell. *Theory of Scheduling*. Reading MA: Addison-Wesley, 1967.

25. Cox, D. R. "A Use of Complex Probabilities in the Theory of Stochastic Processes," *Proceedings of the Cambridge Philosophical Society*, *51*:313–319 (1955).

26. Cox, D. R. and W. Smith. *Queues*. London: Methuen and Co., Ltd, 1961.

27. Crabill, T. B., D. Gross, and M. J. Magazine. "A Classified Bibliography of Research on Optimal Design and Control of Queues," *Operations Research*, *25*:219–232 (1977).

28. Cramér, H. *Mathematical Methods of Statistics*. Princeton: Princeton University Press, 1946.

29. Cumani, A. "On the Canonical Representation of Homogeneous Markov Processes Modelling Failure-Time Distributions," *Microelectronics and Reliability*, *22*(3):583–602 (1982).

30. Dale, A. C. "An Appointments System; Checking Efficiency at Sydenham," *The Hospital*, 569–572 (August 1951).

31. Davis, J. G. and R. Reed, Jr. "Variability Control is the Key to Maximum Operating Room Utilization," *The Modern Hospital*, *102*(4):113–118 (1964).

32. Day, P. W. "Rearrangement Inequalities," *Canadian Journal of Mathematics*, *24*(5):930–943 (1972).

33. Doshi, B. T. "Continuous Time Control of the Arrival Process in an M/G/1 Queue," *Stochastic Processes and Their Applications*, *5*(2):265–285 (1977).

34. Emmons, H. "The Optimal Admission Policy to a Multi-server Queue with Finite Horizon," *Journal of Applied Probability*, *9*:103–116 (1972).

35. Esogbue, A. M. O. "Mathematical and Computational Approaches to some Queueing Processes Arising in Surgery," *Mathematical Biosciences*, *4*(4):531–542 (1969).

36. Fairman, W. L. *Scheduling/Resource Level Decisions in the Hospital Operating Room Environment*. PhD dissertation, University of Pittsburgh, Pittsburgh PA, 1972.

37. Fan, K. "Subadditive Functions on a Distributive Lattice and an Extension of Szász's Inequality," *Journal of Mathematical Analysis Applications*, *18*:262–268 (1967).

38. Fetter, R. and J. Thompson. "Patients' Waiting Time and Doctors' Idle Time in the Outpatient Setting," *Health Services Research*, *1*:66–90 (1966).

39. Finarelli, H. J. *An Algorithm for Scheduling the Hospital Admission of Elective Surgical Patients*. PhD dissertation, University of Pennsylvania, Philadelphia PA, 1971.

40. Fisher, W. C. *Variability in Operating-Room Scheduling at St. Anthony Hospital, Oklahoma City, Oklahoma*. MS thesis, Baylor University, TX, August 1968.

41. Fox, Y. M. "Patient Flow and Scheduling," *Resident and Staff Physician*, *38*:69–76 (1992).

42. French, S. *Sequencing and Scheduling; An Introduction to the Mathematics of the Job-Shop*. Chichester: Horwood, 1982.

43. Frenk, J. B. G. "A General Framework for Stochastic One-Machine Scheduling Problems with Zero Release Times and no Partial Ordering," *Engineering and Informational Sciences*, *5*:297–315 (1991).

44. Fries, B. and V. Marathe. "Determination of Optimal Variables-Sized Multiple-Block Appointment Systems," *Operations Research*, *29*(2):324–345 (1981).

45. Fry, J. "Appointments in General Practice," *Operational Research Quarterly*, *15*(3):233–237 (1964).

46. Gelenbe, E. and G. Pujolle. *Introduction to Queueing Networks*. New York: John Wiley & Sons, 1987.

47. Genkin, A. V.and I. B. Muchnik. "Optimum Algorithm for Maximization of Submodular Functions," *Avtimatica i Telemekhanika*, *8*:139–147 (1990).

48. Glazebrook, K. D. "On Non-Preemptive Strategies for Stochastic Scheduling Problems in Continuous Time," *International Journal of Systems Science*, *12*(6):771–782 (1981).

49. Glazebrook, K. D. "A Suboptimality Bound for Permutation Policies in Single Machine Stochastic Scheduling," *Naval Research Logistics*, *42*:994–1005 (1995).

50. Glazebrook, K. D. and J. C. Gittens. "On Single-Machine Scheduling with Precedence Relations and Linear or Discounted Costs," *Operations Research*, *29*(1):161–173 (1981).

51. Glendenning W. H. *Dental Clinic Scheduling; A Simulation Approach*. MS thesis, AFIT/GSA/SM/72-5, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1972.

52. Goemans, M. X. and V. S. Ramakrishnan. "Minimizing Submodular Functions Over Families of Sets," *Combinatorica*, *15*(4):499–513 (1995).

53. Goldman, J., H. A. Knappenberger, and W. T. Shearon. "Study of the Variability of Surgical Estimates," *Hospital Management*, *110*:46,46A, 46D (1970).

54. Golub, G. H. and C. F. Van Loan. *Matrix Computations*. Baltimore: Johns Hopkins University Press, 1989.

55. Granot, D. and F. Granot. *Optimal Scheduling for an Outpatient Clinic*. Technical Report 137, Austin TX: Center for Cybernetic Studies, University of Texas, April 1973 (AD-769676).

56. Grape, G. R. "Convergence and Cost Minimization in Queuing Systems of the Type (D,M,1)," *FOA (Försvarets Forskningsanstalt) Report*, *2*(1):1–6 (April 1968). Published by the Research Institute of National Defence, Stockholm.

57. Gray, W. J. and P. P. Wang. "On the Computation of Optimal Arrival Times for a Single-Server System." Accepted by *Chinese Institute of Industrial Engineering*, 1994.

58. Häggstrom, O., S. Asmussen, and O. Nerman. *EMPHT - A Program for Fitting Phase-Type Distributions*. Technical Report 1992:4, ISSN 1100-2255, Chalmers University of Technology and The University of Göteborg, 1992.

59. Hall, R. K. *Outpatient Scheduling; A Simulation Approach*. MS thesis, AFIT/GSA/SM/71-3, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1971.

60. Healy, K. J. *Scheduling Arrivals to a Queue*. MS thesis, Pennsylvania State University, University Park PA 16802, May 1982.

61. Healy, K. J., C. D. Pegden, and M. Rosenshine. *Scheduling Arrivals to Multiple Server Queues*. Working Paper 82-128, University Park PA 16802: Department of Industrial and Management Systems Engineering, Pennsylvania State University, May 1982.

62. Heyman, D. "Optimal Operating Policies for M/G/1 Queueing Systems," *Operations Research*, *16*:362–382 (1968).

63. Ho, C. and H. Lau. *Minimizing Total Cost in Scheduling Outpatient Appointments*. Working Paper, Oklahoma State University, 1987.

64. Ho, C. and H. Lau. "Minimizing Total Cost in Scheduling Outpatient Appointments," *Management Science*, *38*(12):1750–1764 (1992).

65. Ho, C., H. Lau, and J. Li. "Introducing Variable-Interval Appointment Scheduling Rules in Service Systems," *International Journal of Operations & Production Management*, *15*(6):58–69 (1995).

66. Hofmann, P. B. and J. F.Rockart. "Implications of the No-Show Rate for Scheduling OPD Appointments," *Hospital Progress*, *50*(8):35–40 (1969).

67. Intel support staff. Telephone conversations. Intel Corp., 7–11 October 1996.

68. Isbell, F. M. *A Statistical Analysis of the Waiting Times in the Air Force Clinics*. MS thesis, Yale University, New Haven, 1963.

69. Jackson, R. R. P. "Design of an Appointments System," *Operational Research Quarterly*, *15*(3):219–224 (1964).

70. Jansson, B. "Choosing A Good Appointment System—A Study of Queues of the Type D/M/1," *Operations Research*, *14*:292–312 (1966).

71. Johansen, S. G. and S. Stidham, Jr. "Control of Arrivals to a Stochastic Input-Output System," *Advances in Applied Probability*, *12*:972–999 (1980).

72. Johnson, M. A. *User's Guide for MEFIT (Version 1.0): A FORTRAN Package for Fitting Mixtures of Erlang Distributions*. Technical Report 90-004, Tucson AZ 85719: The University of Arizona, July 1990.

73. Johnson, M. A. and M. R. Taaffe. "Matching Moments to Phase Distributions: Mixtures of Erlang Distributions of Common Order," *Communications in Statistics - Stochastic Models*, *5*(4):711–743 (1989).

74. Johnson, M. A. and M. R. Taaffe. "A Graphical Investigation of Error Bounds for Moment-Based Queueing Approximations," *Queueing Systems*, *8*(3):295–312 (1991).

75. Johnson, M. A. and M. R. Taaffe. "An Investigation of Phase-Distribution Moment-Matching Algorithms for Use in Queueing Models," *Queueing Systems*, *8*(2):129–147 (1991).

76. Johnson, M. A. and M. R. Taaffe. "Tchebycheff Systems for Probabilistic Analysis," *American Journal of Mathematical and Management Sciences*, *13*(1):83–111 (1993).

77. Johnson, M. A. and Taafe, M. R. "Matching Moments to Phase Distributions: Nonlinear Programming Approaches," *Communications in Statistics - Stochastic Models*, *6*(2):259–281 (1990).

78. Kakalik, J. K. and J. D. C. Little. "Optimal Service Policy for the M/G/1 Queue with Multiple Classes of Arrivals," *RAND Report P4525* (September 1971).

79. Katz, J. "Simulation of Outpatient Appointment Systems," *Communications of the ACM*, *12*:66–90 (1969).

80. Kendall, D. G. "Some Problems in the Theory of Queues," *Journal of the Royal Statistical Society, Series B*, *18*(2):151–185 (1951).

81. Kise, H., T. Ibaraki, and H. Mine. "A Solvable Case of the One-Machine Scheduling Problem with Ready and Due Times," *Operations Research*, *26*(1):121–126 (1978).

82. Kitaev, M. Y. and V. V. Rykov. *Controlled Queueing Systems*. Boca Raton: CRC Press, 1995.

83. Kleinrock, L. *Queueing Systems, Vol I*. New York: John Wiley & Sons, 1975.

84. Kolesar, P. "A Markovian Model for Hospital Admission Scheduling," *Management Science*, *16*(6):B384–B396 (1970).

85. Kuzdrall, P. J., N. K. Kwak, and H. H. Schmitz. "A Technical Note on the Monte Carlo Simulation of Operating-Room and Recovery-Room Usage," *Operations Research*, *22*:434–440 (1974).

86. Kwak, N. K., P. J. Kuzdrall, and H. H. Schmitz. "The GPSS Simulation of Scheduling Policies for Surgical Patients," *Management Science*, *22*:982–989 (1976).

87. Lair, A. V., Mathematics Department Head and M. E. Oxley, Associate Professor of Mathematics. Personal interviews. Air Force Institute of Technology, Wright-Patterson AFB OH, 2–20 February 1997.

88. Lakshminarayan, S., R. Lakshmanar, R. L. Panineau, and R. Rochette. "Optimal Single-Machine Scheduling with Earliness and Tardiness Penalties," *Operations Research*, *26*(6):1079–1082 (1978).

89. Lang, A. *Parameter Estimation for Phase-Type Distributions, Part I: Fundamentals and Existing Methods*. Technical Report 159, Corvallis OR 97331-4606: Department of Statistics, Oregon State University, February 1994.

90. Lang, A. *Parameter Estimation for Phase-Type Distributions, Part II: Computational Evaluation.* Technical Report 160, Corvallis OR 97331-4606: Department of Statistics, Oregon State University, August 1994.

91. Lawler, E. L. "Sequencing to Minimize the Weighted Number of Tardy Jobs," *Revue Francaise d'Automatique, d'Informatique, et de Recherche Operationelle, 10.5 Suppl.*:27–33 (1976).

92. Lawler, E. L. "A 'Pseudopolynomial' Time Algorithm for Sequencing Jobs to Minimize Total Tardiness," *Annals of Discrete Mathematics, 1*(2):331–342 (1977).

93. Lawler, E. L., J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *Sequencing and Scheduling: Algorithms and Complexity.* Technical Report BS-R89xx, Center for Mathematics and Computer Science, Amsterdam, 1989.

94. Lee, H.and G. L. Nemhauser and Y. Wang. "Maximizing a Submodular Function by Integer Programming: Polyhedral Results for the Quadratic Case," *European Journal of Operational Research, 94*:154–166 (1996).

95. Liao, C. *Planning Timely Arrivals to a Stochastic Production or Service System.* PhD dissertation, Pennsylvania State University, University Park PA 16802, August 1988.

96. Liao, C., C. D. Pegden, and M. Rosenshine. *Planning Timely Arrivals to a Stochastic Production or Service System.* Working Paper 88-136, University Park PA 16802: Department of Industrial and Management Systems Engineering, Pennsylvania State University, 1988.

97. Liao, C. and C. D. Pegden and M. Rosenshine. "Planning Timely Arrivals to a Stochastic Production or Service System," *IIE Transactions, 25*(5):63–73 (1993).

98. Lin, Y. Y. and L. P. Hwang. "Efficient Computation of the Matrix Exponential Using Padé Approximation," *Computers in Chemistry, 16*(4):285–293 (1992).

99. Lindley, D. V. "The Theory of Queues with a Single Server," *Proceedings of the Cambridge Philosophical Society, 48*:277–289 (1952).

100. Liou, M. L. "A Novel Method of Evaluating Transient Response," *Proceedings of the IEEE, 54*(1):20–23 (1966).

101. Lorentz, G. G. "An Inequality for Rearrangements," *American Mathematics Monthly, 60*(3):176–179 (1953).

102. Magerlein, J. M. and J. B. Martin. "Surgical Demand Scheduling: A Review," *Health Services Research, 13*:418–433 (1978).

103. Marshall, A. W. and I. Olkin. *Inequalities: Theory of Majorization and Its Applications.* New York: Academic Press, 1979.

104. Mercer, A. "A Queueing Problem in Which the Arrival Times of the Customers are Scheduled," *Journal of the Royal Statistical Society, Series B*, *22*(1):108–113 (1960).

105. Mercer, A. "Queues with Scheduled Arrivals: A Correction, Simplification, and Extension," *Journal of the Royal Statistical Society, Series B*, *35*(1):104–116 (1973).

106. Microsoft support staff. Telephone conversations. Microsoft Corp., Seattle WA, 16–23 September 1996.

107. Möhring, R. H., F. J. Radermacher, and G. Weiss. "Stochastic Scheduling Problems I: General Strategies," *Zeitschrift für Operations Research*, *28*:193–260 (1984).

108. Moler, C. and C. Van Loan. "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review*, *20*(4):801–837 (1978).

109. Moler, Cleve. Personal correspondence. Mathworks Company, Natick MA, 21–23 September 1996.

110. Moré, J. J.and Zhijun Wu. "Global Continuation for Distance Geometry Problems," *Siam Journal on Optimization*, 7(3):814–836 (1997).

111. Morse, P. M. *Queues, Inventories, and Maintenance; The Analysis of Operational Systems With Variable Demand and Supply*. New York: John Wiley & Sons, 1963.

112. Muth, E. J. "The Effect of Uncertainty in Job Times on Optimal Schedules." *Industrial Scheduling* edited by Muth, J. F. and G. L. Thompson, Englewood Cliffs: Prentice-Hall, 1963.

113. Naor, P. "The Regulation of Queue Size by Levying Tolls," *Econometrica*, *37*(1):15–23 (1969).

114. Nardino, R., Maj, USAF, doctor at Wright-Patterson Air Force Base Primary Care Clinic. Discussions, April 1997.

115. Nash, P. and R. R. Weber. "Dominant Strategies in Stochastic Allocation and Scheduling Problems." *Deterministic and Stochastic Scheduling* edited by Dempster, M.A.H., et al., 343–353, Dordrecht, Holland: D. Reidel, 1982. Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, Durham, England, in 1981.

116. Nelder, J. A. and R. Mead. "A Simplex Method for Function Minimization," *Computer Journal*, 7:308–313 (1964).

117. Neuts, M. F. *Matrix-Geometric Solutions in Stochastic Models; An Algorithmic Approach*. New York: Dover, 1981.

118. Neuts, M. F. *Phase-Type Distributions: A Bibliography*. Working Paper, Tucson AZ 85721: The University of Arizona, 1988.

119. Newman, D. J. and Reddy, A. R. "Rational Approximations to e^x and to Related Functions," *Journal of Approximation Theory*, *25*:21–30 (1979).

120. Nuffield Provincial Hospitals Trust. *Studies in the Functions and Design of Hospitals*. London: Oxford University Press, 1955.

121. O'Cinneide, C. "On Non-Uniqueness of Representations of Phase-Type Distributions," *Communications in Statistics - Stochastic Models*, *5*(2):247–259 (1989).

122. O'Keefe, R. "Investigating Outpatient Departments: Implementable Policies and Qualitative Approaches," *Journal of the Operational Research Society*, *36*(8):705–712 (1985).

123. Parlett, B. N. *Computation of Functions of Triangular Matrices*. Working Paper ERL-M481, Electronics Research Laboratory, College of Engineering, University of California, Berkeley CA 94720: University of California, Berkeley, 1974.

124. Parlett, R. N. "A Recurrence Among the Elements of Functions of Triangular Matrices," *Linear Algebra and Its Applications*, *14*(2):117–121 (1976).

125. Pegden, C. D. and M. Rosenshine. *Scheduling Arrivals to Queues*. Working Paper 82-101, University Park PA 16802: Department of Industrial and Management Systems Engineering, Pennsylvania State University, January 1982.

126. Pegden, C. D. and M. Rosenshine. *Scheduling Queueing Arrivals into Time Slots*. Working Paper 83-136, University Park PA 16802: Department of Industrial and Management Systems Engineering, Pennsylvania State University, 1983.

127. Pegden, C. D. and M. Rosenshine. "Scheduling Arrivals to Queues," *Computers in Operations Research*, *17*(4):343–348 (1990).

128. Philips, K. T. "Operating Room Utilization," *Hospital Topics*, *53*:42–45 (1975).

129. Pinedo, M. "Stochastic Scheduling with Release Dates and Due Dates," *Operations Research*, *31*(3):559–572 (1983).

130. Pinedo, M. *Scheduling; Theory, Algorithms, and Systems*. Englewood Cliffs: Prentice Hall, 1995.

131. Press, W. H., B. P. Flannery, S. A. Teulkolsky, and W. T. Vetterling. *Numerical Recipes; The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1987.

132. Przasnyski, Z. H. "Operating Room Scheduling," *AORN Journal*, *44*(1):67–79 (July 1986).

133. Righter, R. "Scheduling." *Stochastic Orders and Their Applications* edited by Shaked, M. and J.G. Shantikumar, Boston: Academic Press, 1994.

134. Rockafellar, R. T. *Convex Analysis*. Princeton: Princeton University Press, 1970.

135. Rothkopf, M. H. "Scheduling With Random Service Times," *Management Sciences*, *12*(9):707–712 (1966).

136. Rothkopf, M. H. and S. A. Smith. "There are no Undiscovered Priority Index Sequencing Rules for Minimizing Total Delay Costs," *Operations Research*, *32*(2):451–456 (1984).

137. Sabria, F. and C. F. Daganzo. "Approximate Extensions for Queueing Systems with Scheduled Arrivals and Established Service Order," *Transportation Science*, *23*(3):159–165 (1989).

138. Schmickler, L. "MEDA: Mixed Erlang Distributions as Phase-Type Representations of Empirical Distribution Functions," *Communications in Statistics - Stochastic Models*, *8*(1):259–281 (1992).

139. Schmitz, H. H. and N. K. Kwak. "Monte Carlo Simulation of Operating-Room and Recovery-Room Usage," *Operations Research*, *20*:1171–1180 (1972).

140. Schroer, B. J. and H. T. Smith. "Effective Patient Scheduling," *The Journal of Family Practice*, *5*(3):407–411 (September 1977).

141. Schwimer, J. "On the N-Job, One-Machine, Sequence-Independent Scheduling Problem With Tardiness Penalties: A Branch-Bound Solution," *Management Sciences*, *18*(6):B-310–B-313 (1972).

142. Sevcik, K. C. "Scheduling for Minimum Total Loss Using Service Time Distributions," *Journal of the Association for Computing Machinery*, *21*(1):66–75 (1974).

143. Shantikumar, J. G. and D. D. Yao. "Second-Order Stochastic Properties in Queueing Systems," *Proceedings of the IEEE*, *77*(1):162–170 (1989).

144. Sidney, J. B. "Optimal Single-Machine Scheduling with Earliness and Tardiness Penalties," *Operations Research*, *25*(1):62–69 (1977).

145. Simeoni, J. R. *An Efficient Approach to Solving the Control of Arrivals Problem*. MS thesis, AFIT/GOR/ENS/94M-14, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1994.

146. Sisson, R. L. "Sequencing Theory." *Progress in Operations Research, Vol I* edited by Ackoff, R. L., 295–326, New York: John Wiley & Sons, 1963.

147. Smith, C. M. and B. P. Yawn. "Factors Associated with Appointment Keeping in a Family Practice Residency Clinic," *The Journal of Family Practice*, *38*(1):25–29 (1994).

148. Sobel, M. J. "Optimal Operation of Queues." *Mathematical Methods in Queueing Theory; Proceedings of a Conference at Western Michigan University, May 10-12, 1973* edited by Clarke, A.B., 231–262, Berlin: Springer-Verlag, 1974.

149. Soriano, A. "Comparison of Two Scheduling Systems," *Operations Research*, *14*:388–397 (1966).

150. Soriano, A. "On the Problem of Batch Arrivals and its Application to a Scheduling System," *Operations Research*, *14*:398–408 (1966).

151. Standish, C. J. "Truncated Taylor Series Approximation to the State Transition Matrix of a Continuous Parameter Finite Markov Chain," *Linear Algebra and Its Applications*, *12*(2):179–183 (1975).

152. Stidham, S., Jr. "Optimal Control of Admission to a Queueing System," *IEEE Transactions on Automatic Control*, *AC-30*(8):705–713 (1985).

153. Stidham, S., Jr. and N. U. Prabhu. "Optimal Control of Queueing Systems." *Mathematical Methods in Queueing Theory; Proceedings of a Conference at Western Michigan University on May 10-12, 1973* edited by A.B. Clarke, 263–294, Berlin: Springer-Verlag, 1974.

154. Tijms, H. *Stochastic Models; An Algorithmic Approach*. Chichester: John Wiley & Sons, 1994.

155. Topkis, D. M. *Ordered Optimal Solutions*. PhD dissertation, Stanford University, Stanford CA, 1968.

156. Topkis, D. M. "Minimizing a Submodular Function on a Lattice," *Operations Research*, *26*(2):305–321 (1978).

157. Topkis, D. M. Personal correspondence. University of California-Davis, 8 June 1997.

158. Vanden Bosch, P. M., D. C. Dietz, and J. R. Simeoni. *Scheduling Customer Arrivals for a Stochastic Service System*. Working Paper WP97-1, Department of Operational Sciences, Air Force Institute of Technology, April 1997.

159. Voorhis, W. R. "Waiting-Line Theory as a Management Tool," *Operations Research*, *4*(6):221–228 (1956). Edited by R. L. Ackhoff.

160. Wang, P. P. "Static and Dynamic Scheduling of Customer Arrivals to a Single-Server System," *Naval Research Logistics*, *40*:345–60 (1993).

161. Wang, P. P. "Releasing N Jobs to an Unreliable Machine," *Computers in Industrial Engineering*, *26*(4):661–671 (1994).

162. Wang, P. P. Personal correspondence, June 1996.

163. Wang, P. P. "Optimally Scheduling N Customer Arrival Times for a Single-Server System," *Computers in Operations Research*, *24*(8):703–716 (1997).

164. Weiss, E. N. "Models for Determining Estimated Start Times and Case Ordering in Hospital Operating Rooms," *IIE Transactions*, *22*(2):143–150 (1990).

165. Weiss, G. "A Tutorial in Stochastic Scheduling." *Scheduling Theory and its Applications* edited by Chrétienne, P., et al., New York: John Wiley & Sons, 1995.

166. Welch, J. D. "Appointment Systems in Hospital Outpatient Departments," *Lancet*, 1105–1109 (31 May 1952).

167. Welch, J. D. "Appointments Systems in Hospital Outpatient Departments," *Operational Research Quarterly*, *15*(3):224–231 (1964).

168. White, M. and M. Pike. "Appointment Systems in Outpatient's Clinics and the Effect of Patients' Unpunctuality," *Medical Care*, *2*:133–145 (1964).

169. Whitt, W. "Approximating a Point Process by a Renewal Process, I: Two Basic Methods," *Operations Research*, *30*(1):125–147 (1982).

170. Wilks, S. *Mathematical Statistics*. New York: John Wiley & Sons, 1962.

171. Winsten, C. B. "Geometric Distribution in the Theory of Queues," *Journal of the Royal Statistical Society, Series B*, *21*:1–35 (1959).

172. Yechiali, U. "Customers' Optimal Joining Rules for the GI/M/s Queue," *Management Science*, *18*(7):434–443 (1985).

Vita

Major Peter Vanden Bosch was graduated from Michigan State University Honors College with a Bachelor of Arts degree in physics in 1977 and from Saint Mary's University, San Antonio, TX, in 1988 with a Master of Science degree in operations research.

He taught high school mathematics, physics, and computer science in Chicago and Detroit until 1984, when he accepted a commission in the Air Force. Analytical USAF assignments include: basic research in nuclear fallout transport, microwave bioeffects, and statistics at the USAF School of Aerospace Medicine, Brooks AFB, TX; application of expert systems control to manufacturing processes at the Materials Laboratory at Wright-Patterson AFB, OH; and air-to-air combat analysis and modeling for ACC Studies and Analyses, Langley AFB, VA. Other USAF assignments include missile launch officer, financial resource manager, and program manager. After completion of his PhD at the Air Force Institute of Technology, he will become an analyst for the Air Staff Personnel Deputate at the Pentagon.

Publications not referred to in this dissertation are: "Radiation Doses From Flying Through Nuclear Debris Clouds," USAF School of Aerospace Medicine Technical Report, 1985; "Optimization of Temperature Distribution in a Dispersive Slab Exposed to Radiofrequency Radiation", Master's Thesis, 1987; and "A Singular Function," *The Mathematics Teacher*, May 1997.

Electronic correspondence may be directed to: pvandenb@compuserve.com

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1997	PhD dissertation

4. TITLE AND SUBTITLE

Scheduling and Sequencing Arrivals to a Stochastic Service System

5. FUNDING NUMBERS

6. AUTHOR(S)

Vanden Bosch, Peter M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT/ENS
2950 P Street
Wright-Patterson AFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DS/ENS/97-03

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Capt Rick Jenkins
Human Resources Directorate
Manpower and Personnel Research Division
7909 Lindbergh Drive
Brooks AFB, TX 78235-5352

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Optimization of scheduled arrival times to an appointment system is approached from the perspectives of both queueing and scheduling theory. The appointment system is modeled as a one-server, first-come-first-served, transient queue with independent, distinctly distributed service times and no-show rates. If a customer does show, it is assumed to be punctual. The cost of operating the appointment system is a convex combination of customers' waiting times and the server's overtime. While techniques for finding the optimal static and dynamic schedules of arrivals have been proposed by other researchers, they mainly have focused on identical customers and strictly punctual arrivals. This effort provides substantially more efficient solution methods, addresses a more general cost function, allows for no-shows and non-identical service distributions, and applies either when arrivals are constrained to lattice points or when they are unconstrained. Because customers are not indistinguishable, this effort also provides heuristics for determining optimal customer order. The effort concentrates on medical scheduling examples but is applicable to any appointment scheduling operation. Further, the proposed techniques apply to any convex, submodular function.

14. SUBJECT TERMS
Appointment, stochastic service system, scheduling, sequencing, moment matching

15. NUMBER OF PAGES
267

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL